

## **4.12 Product Distribution System (PDS) Subsystem Overview**

Deleted

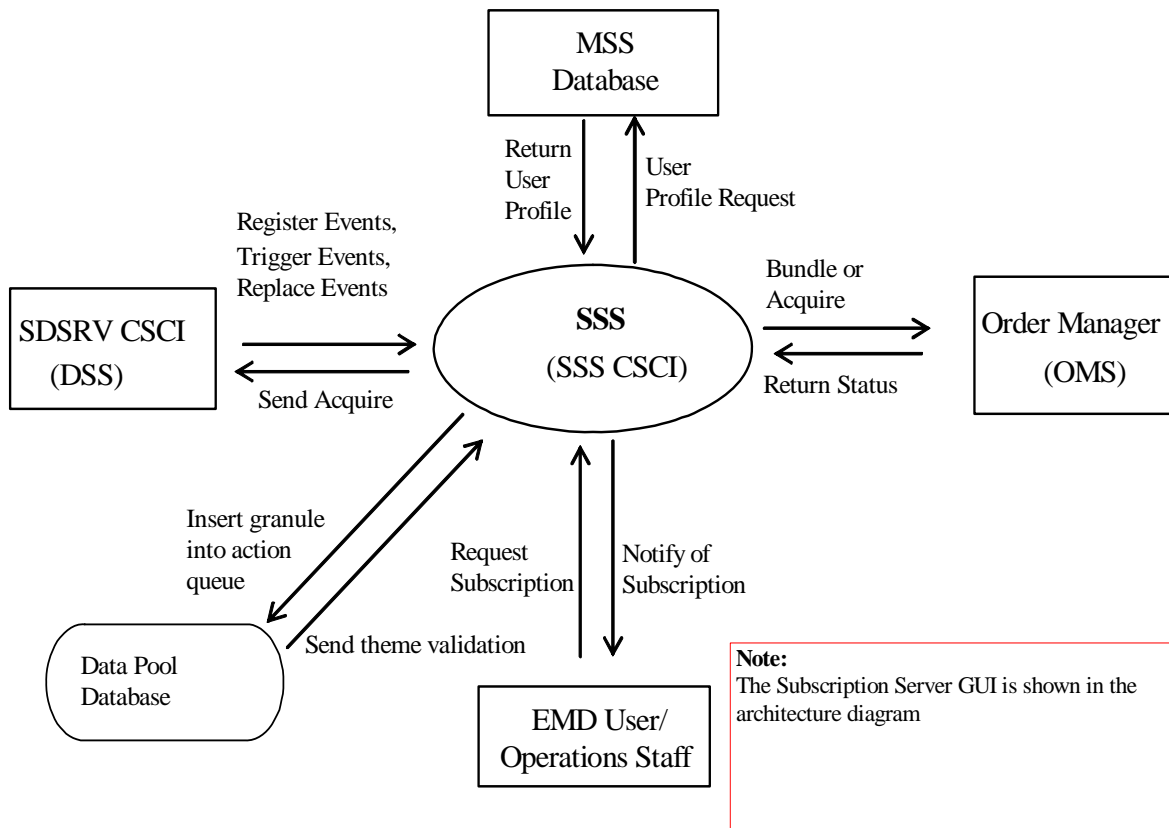
### 4.13 Spatial Subscription Server (SSS) Subsystem Overview

The Spatial Subscription Server (SSS) subsystem is the principal means by which users can establish standing orders for data. Users enter subscriptions for specific ESDTs using a GUI or command line interface (CLI). A subscription may be qualified by specifying one or more constraints on the metadata of matching granules. This includes the capability of qualifying the subscription spatially by specifying a geographic area (rectangle) over which the data was collected. A subscription has one or more associated actions such as data distribution, email notification, Data Pool insertion, or bundling, i.e. adding a granule to an Order Manager bundle.

In addition to the subscription creation components, the SSS subsystem is comprised of a database, installed on a Sybase ASE server, and four runtime drivers: an event driver to match subscriptions with granule events, an action driver to execute the actions of matched subscriptions, a recovery driver to restart stalled events or actions, and a deletion driver to clean up the database.

#### Spatial Subscription Server (SSS) Context

Figure 4.13-1 is the Spatial Subscription Server context diagram. Table 4.13-1 provides descriptions of the interface events in the Spatial Subscription Server context diagram.



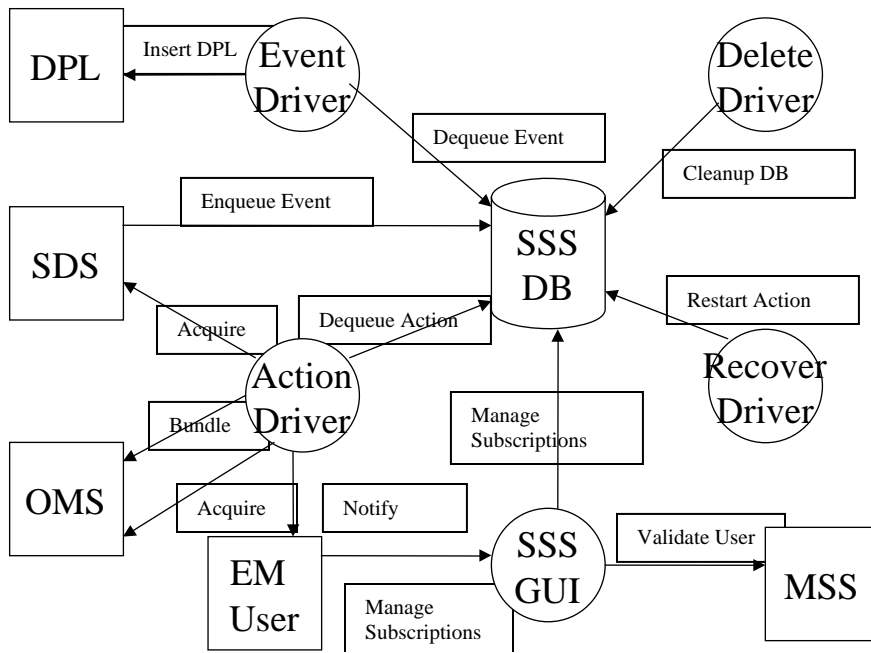
**Figure 4.13-1. Spatial Subscription Server Context Diagram**

**Table 4.13-1. Subscription Server Interface Events**

<b>Event</b>	<b>Interface Event Description</b>
User Profile Request	<b>User Profile Request</b> - MSS provides requesting CSCIs with User Profile parameters such as e-mail address and shipping address to support their processing activities. SSS will not enter a subscription for or distribute a granule to a user who does not have a valid MSS user profile.
Notify of Subscription	The SSS CSC sends email notification to the <b>EMD User</b> when the subscribed event occurs, provided that a notification action was requested in the subscription.
Request Subscription	A subscriber ( <b>EMD user</b> requests <b>Operations Staff</b> to create the subscription) sends information (ESDT and, optionally, acceptable metadata values) with the subscription, specifying one or more actions (e.g., acquire and/or notification) to be taken when the subscribed event occurs.
Send Acquire	An “acquire” (instructions to obtain data) is sent to the <b>SDSRV CSCI</b> via its command line interface, provided that this option is set in the SSS Action Driver configuration parameters (alternatively, acquires may be sent to OMS).
Return status	Status returned by a stored procedure to indicate whether or not the call succeeded.
Register Events	The <b>SDSRV CSCI</b> sends information about an Earth Science Data Type (ESDT) to the SSS database when an ESDT is installed into the system.
Trigger Events	The <b>SDSRV CSCI</b> notifies the SSS database (via a database trigger) when a granule arrives matching the ESDT of at least one subscription.
Replace Events	The <b>SDSRV CSCI</b> notifies the SSS database when an ESDT is deleted from the system.
Return User Profile	MSS returns the user profile to SSS as part of user authentication.
Bundle or Acquire	SSS notifies OMS, via a stored procedure call, when a granule has matched a subscription. If the subscription is bundled, i.e. associated with an OMS bundling order, then the granule is inserted into the appropriate OMS bundle. If the subscription is not bundled, then an acquire request is sent to OMS. Whether an acquire is sent to SDS or OMS is determined by an SSS configuration parameter.
Insert granule into action queue	If a subscription has an associated Data Pool action, then SSS will insert a row into the Data Pool database action queue table, indicating that the granule that matched the subscription should be inserted into the Data Pool.
Send theme validation	If a subscription's Data Pool action is associated with a Data Pool theme, then the Data Pool will verify, via stored procedure call that the theme exists and is enabled for insert.

### **4.13.1 Spatial Subscription Server Architecture**

Figure 4.13-2 is the Spatial Subscription Server architecture diagram. The diagram shows the events sent to the Spatial Subscription Server processes and the events the Subscription Server processes send to other processes.



**Figure 4.13-2. Spatial Subscription Server Architecture Diagram**

Table 4.13-2 provides descriptions of the processes shown in the Spatial Subscription Server architecture diagram.

**Table 4.13-2. Spatial Subscription Server Processes**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcNbSubscribedEventDriver ("Event Driver")	Server	ACMHW	Developed	The SSS event driver dequeues events and matches them with active subscriptions. Information about matched subscriptions is placed in the action queue. If a matched subscription has a Data Pool action, the event driver inserts information into the Data Pool database.
EcNbActionDriver ("Action Driver")	Server	ACMHW	Developed	The SSS action driver dequeues matched subscriptions and executes their associated actions (acquire or notification). An acquire may be directed either to the Science Data Server or to the Order Manager, according to the value of a configuration parameter. If a subscription is bundled, then the granule that matched it is added to that bundle via an OMS interface.
EcNbDeleteRequestDriver ("Delete Driver")	Server	ACMHW	Developed	The SSS delete driver dequeues from the delete request queue and cleans up database storage for the completed action or event.
EcNbRecoverDriver ("Recover Driver")	Server	ACMHW	Developed	The SSS recover driver monitors the event and action queues for stalled events/actions and reenqueues them so that they will be tried again.
EcNbSubscriptionGUI ("SSS GUI")	GUI	DPSHW	Developed	The SSS GUI provides an operator interface for submitting, updating and deleting subscriptions. It is also used for creating OMS bundling orders and for bundling subscriptions to bundling orders.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE is where the SSS database resides.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

#### **4.13.1.1 Subscription Server Process Interface Descriptions**

Table 4.13-3 provides descriptions of the interface events shown in the Subscription Server architecture diagram.

**Table 4.13-3. Spatial Subscription Server Process Interface Events (1 of 3)**

Event	Event Frequency	Interface	Initiated by	Event Description
Enqueue Event	Once per granule ingest	<i>Process:</i> TrigInsDsBtNotifierQueue	<i>Process:</i> Insert into table DsBtNotifierQueue	When a new granule is ingested, a row is inserted into the DsBtNotifierQueue table in the Science Data Server database. An insert trigger on this table, checks to see if there is at least one subscription in the SSS database for the same collection type. If so, the trigger inserts a row into the SSS event queue, i.e. the table EcNbSubscribedEventQueue in the SSS database.
Dequeue Event	Once per event	<i>Process:</i> ProcSubscribedEventDequeue	<i>Process:</i> EcNbSubscribedEventDriver	An event driver instance will dequeue up to 10 events from the event queue at one time. It will then process the events sequentially by getting the metadata for each granule and comparing it with the list of active subscriptions. If a subscription matches a granule event, information about the match is placed into the action queue.
Insert DPL	Once per event	<i>Process:</i> TrigInsEcNbDpEventDetails	<i>Process:</i> EcNbSubscribedEventDriver	When a granule event matches one or more subscriptions, at least one of which has an associated Data Pool action, the event driver will insert information about the granule (with subscription numbers) into the Data Pool database. A single insert per event is performed by an insert trigger on the table EcNbDpEventDetails.

**Table 4.13-3. Spatial Subscription Server Process Interface Events (2 of 3)**

Event	Event Frequency	Interface	Initiated by	Event Description
Dequeue Action	Once per matched subscription	<i>Process:</i> ProcActionDequeue	<i>Process:</i> EcNbActionDriver	An action driver instance will dequeue up to 10 matched subscriptions from the action queue at one time. It will then process them sequentially by getting the actions for each subscription. If the subscription is bundled, then the granule is added to the current bundle for that bundling order via a stored procedure call to the OMS database. Otherwise, the action driver will initiate an acquire of the granule (using either SDS or OMS, depending on the configuration) or send email notification to the user, depending on how the subscription was set up.
Acquire	Once per matched subscription	<i>Process:</i> OmCreateNonBundlingOrder (OMS case) SCLI acquire (SDS case)	<i>Process:</i> EcNbActionDriver	If a matched subscription has an associated acquire action, the action driver will initiate the acquire, either by (1) stored procedure call to OMS, or (2) the SDS command line interface. Which method is used depends upon the configuration setting.
Bundle	Once per matched subscription	<i>Process:</i> OmInsertBundleRequest	<i>Process:</i> EcNbActionDriver	If a matched subscription is a bundled subscription, the action driver will send information about the granule to OMS via a stored procedure call.
Notify	Once per matched subscription	<i>Process:</i> mailx	<i>Process:</i> EcNbActionDriver	If a matched subscription has an associated notification action, the action driver will compose an email message and send it to the address specified in the subscription definition.

**Table 4.13-3. Spatial Subscription Server Process Interface Events (3 of 3)**

Event	Event Frequency	Interface	Initiated by	Event Description
Restart Action	Once per action or event	<i>Process:</i> ProcActionReEnqueue, ProcSubscribedEventReEnqueue	<i>Process:</i> EcNbRecoverDriver	If an action or event appears to have stalled, i.e. did not run to completion based on evidence in the log tables, the recover driver will reenqueue the action or event in its appropriate queue.
Cleanup DB	Once per action or event	<i>Process:</i> ProcDequeueDeleteRequest, ProcDeleteProcessedSub, ProcDeleteProcessedEvent	<i>Process:</i> EcNbDeleteRequestDriver	The delete driver will clean up tables in the database based on entries in the delete request queue. Each entry in this queue corresponds to one action or one event.
Manage Subscriptions	Various	<i>Process:</i> EcNbSubscriptionGUI	<i>Process:</i> EcNbSubscriptionGUI	The SSS GUI allows a user to create, delete, edit or view subscriptions. Or to create, delete, edit or view bundling orders and bundle subscriptions to them.
Validate User	Once per subscription creation	<i>Process:</i> EcNbSubscriptionGUI	<i>Process:</i> EcNbSubscriptionGUI	The SSS GUI will verify that any subscription owner or recipient of a granule distribution has a valid user profile present in the MSS database.



#### 4.13.1.2 Subscription Server Data Stores

Spatial Subscription Server uses the COTS software Sybase Adaptive Server Enterprise (ASE) for the storage of persistent data. The following is a brief description of the principal types of data contained in the database:

- **Attributes:** includes the ESDTs for which subscriptions can be created and the metadata attributes that can be used to qualify those subscriptions.
- **Subscriptions:** information about subscriptions that have been created for users, their associated qualifying expressions, and their associated actions.
- **Events:** information about newly arrived data granules, their metadata, and the subscriptions that match them.
- **Actions:** information about actions for matched subscriptions that need to be carried out, e.g. acquire or email notification.

Table 4.13-4 provides descriptions of the data found in the principal Sybase ASE data stores used by the Spatial Subscription Server. More detail on these and other data stores can be found in the Spatial Subscription Server Database Design and Schema Specifications for the EMD Project (Refer to CDRL 311).

**Table 4.13-4. Spatial Subscription Server Data Stores (1 of 2)**

Data Store	Type	Functionality
EcNbEventDefinition	Attributes	Contains the list of events to which a user can subscribe.
EcNbEventMetadataAttrDef	Attributes	Contains the list of attributes which can be used to qualify a subscription.
EcNbEventAttrXref	Attributes	Cross-references subscribable events with the metadata attributes pertaining to them.
EcNbSubscription	Subscriptions	Contains the list of user subscriptions.
EcNbMatchingExpression	Subscriptions	Contains the list of expressions used to qualify subscriptions.
EcNbSubMatchExp_XREF	Subscriptions	Cross-references subscriptions with matching expressions (qualifiers).
EcNbSubMatchingExpInteger	Subscriptions	Contains the range of integer values used to qualify a subscription by an integer attribute.
EcNbSubMatchingExpFloat	Subscriptions	Contains the range of float values used to qualify a subscription by a float attribute.
EcNbSubMatchingExpString	Subscriptions	Contains the string values used to qualify a subscription by a string attribute.
EcNbSubMatchingExpDate	Subscriptions	Contains the range of date values used to qualify a subscription by a date attribute.
EcNbNoseMatchingExpression	Subscriptions	Contains the values used to qualify a subscription by orbit data.
EcNbSpatialMatchingExpression	Subscriptions	Contains the values used to qualify a subscription spatially.
EcNbActionDefinition	Subscriptions	Contains the list of actions associated with subscriptions.

**Table 4.13-4. Spatial Subscription Server Data Stores (2 of 2)**

<b>Data Store</b>	<b>Type</b>	<b>Functionality</b>
EcNbOrderAction	Subscriptions	Contains detailed information about acquire actions associated with subscriptions.
EcNbNotificationAction	Subscriptions	Contains detailed information about email notification actions associated with subscriptions.
EcNbDpAction	Subscriptions	Contains detailed information about Data Pool actions associated with subscriptions.
EcNbSubscribedEventQueue	Events	Contains information about granules which have entered the system that could match user subscriptions.
EcNbSubEventQueueLog	Events	A log of all operations performed on the subscribed event queue.
EcNbEventMetadataInteger	Events	Contains metadata values for integer attributes of granule events.
EcNbEventMetadataFloat	Events	Contains metadata values for float attributes of granule events.
EcNbEventMetadataString	Events	Contains metadata values for string attributes of granule events.
EcNbEventMetadataDate	Events	Contains metadata values for date attributes of granule events.
EcNbEventMetadataNose	Events	Contains metadata values for attributes of granule events relating to orbit data.
EcNbDpEventDetails	Events	Used by the event driver to process Data Pool actions.
EcNbEventTruth	Events	Used by the event driver as part of the matching algorithm between granule events and user subscriptions.
EcNbActionQueue	Actions	Contains information about subscriptions which have been matched with granule events.
EcNbActionQueueLog	Actions	A log of all operations performed on the action queue.
EcNbDistribution	Actions	Used by the action driver to suppress duplicate distribution of granules.
EcNbDeleteRequestQueue	Events, Actions	A list of actions and events that can be removed from the database.

## 4.14 Data Pool Subsystem Overview

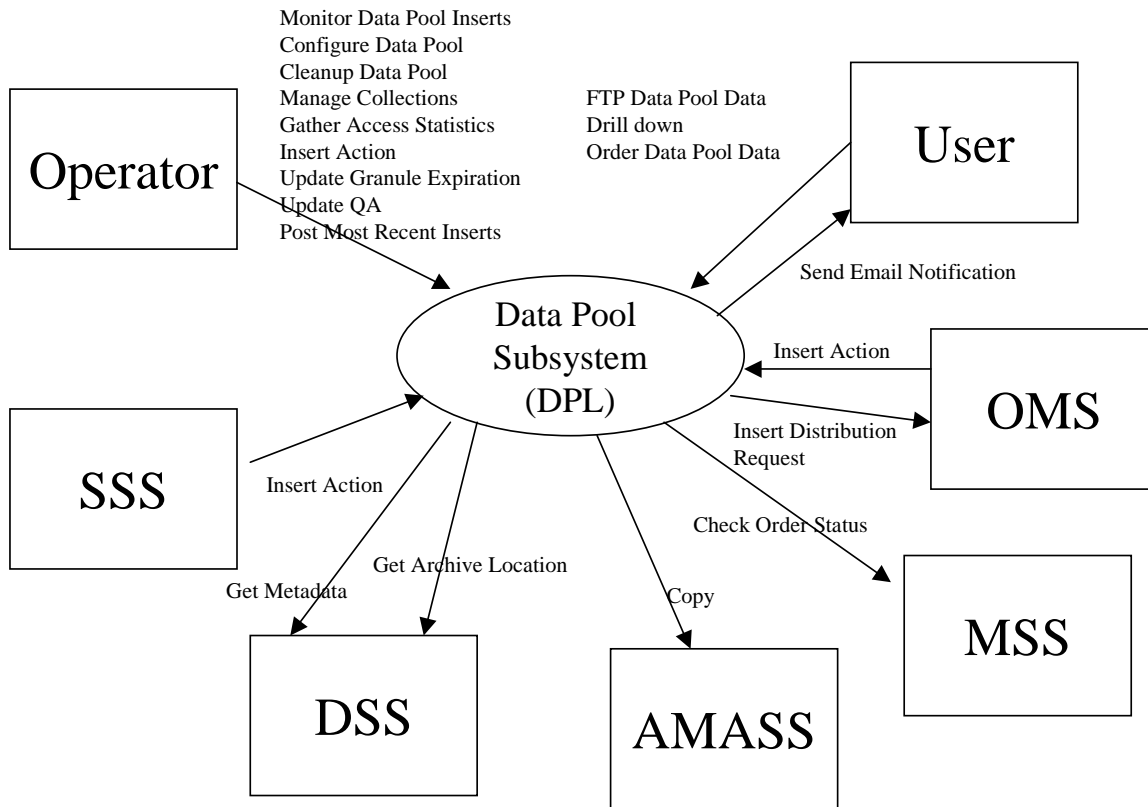
The Data Pool is a large online cache of data at each DAAC. Data in the Data Pool may be either data from the ECS archive, or non-ECS data selected by the DAAC. Science, metadata (in xml format), and browse files (in jpg format) are stored in the Data Pool. The Data Pool subsystem consists of the following components and supporting utilities:

1. **Data Pool Insert:** inserts ECS or non-ECS data into the Data Pool. EMD data is copied from AMASS into the Data Pool, based on either Data Pool insert actions assigned by the Spatial Subscription Server or Order Manager Server, or on EMD granule ids submitted by the Batch Insert Utility. Non-ECS data is copied into the Data Pool from a location specified in an xml file created by DAAC operations as input to the Batch Insert Utility. The Data Pool database inventory is updated for each granule inserted in the Data Pool. Data Pool Insert consists of six major subcomponents: a) the Data Pool Action Driver (EcDIActionDriver) (DPAD): a C++ executable which schedules Data Pool insert actions based on a queue of Data Pool insert actions populated by the Spatial Subscription Server, the Batch Insert Utility, or the Order Manager Server; b) the Data Pool Insert Utility (EcDIInsertUtility) (DPIU), a java executable which manages the insert of a data granule into the Data Pool, and which updates the Data Pool database inventory and writes xml metadata files to the Data Pool disk; c) the Data Pool Quick Server (EcDIQuickServer), a C++ executable which is installed on each AMASS host and which checks AMASS file status and performs the actual copy operations from AMASS to the Data Pool for EMD data using the EcUtCopyExec; d) the Data Pool XML generator (EcDIM2XTApp), a java executable which translates EMD granule metadata from the Science Data Server database into XML, for storage in the Data Pool directories; e) the band extraction utility (bandtool), a C executable invoked by the DPIU, which extracts band information from HDF-EOS granules and stores the extracted information in the Data Pool database. The bandtool is invoked only if the granule being inserted is from a collection eligible for conversion by the HDF-EOS to GeoTiff Conversion Tool (HEG); and f) the jpeg extraction utility (hdf2jpeg), a C executable invoked by the DPIU, which extracts browse images (jpeg or raster) from a browse hdfs file.
2. **Data Pool Cleanup and Validation (EcDICleanupDataPool.pl):** a perl utility, which cleans expired granules from the Data Pool directories and database. This utility normally runs as a cron job. The utility may also be used to report on and correct inconsistencies between the Data Pool directories and the database (validation).
3. **Data Pool Web Access (EcDIWebAccess):** a java-based web application, which runs with the apache web server and related COTS. The Data Pool Web Access application allows end-users to perform drill-down searches for Data Pool data, to view metadata and browse images online, and to convert and/or order Data Pool data.
4. **Data Pool Maintenance GUI (EcDIDpm):** a perl-based web GUI that allows DAAC operations staff to monitor Data Pool insert activity and to control the Data Pool configuration.

5. **Data Pool Access Statistics utilities:** perl utilities which parse firewall ftp logs (EcDIRollupFwFtpLogs.pl) and Data Pool Web Access custom code logs (EcDIRollupWebLogs.pl) for accesses to the Data Pool directories, and then roll up access information for storage in the Data Pool database.
6. **Data Pool FTP Server:** customized wu-ftp daemon, which supports ftp access to Data Pool directories and also provides a checksum-on-download service.
7. **Data Pool Update Granule Expiration utility (EcDIUpdateGranule.pl):** a perl utility, which allows operations staff to update the Data Pool expiration date and retention priority for specified Data Pool granules.
8. **Data Pool Batch Insert Utility (EcDIBatchInsert.pl):** a perl utility, which allows operations staff to queue ECS or non-ECS data for insert into the Data Pool.
9. **Data Pool Most Recent Insert Utility (EcDIMostRecentInsert.pl):** a perl utility, which creates files at the file system and data collection level of the Data Pool directory structure which contain information about granules recently inserted into the Data Pool at those levels.
10. **Data Pool Collection Remapping Utility (EcDIRemap.pl):** a perl utility, which allows DAAC operations staff to remap all data in a Data Pool collection directory from one higher level collection group directory to another.
11. **Data Pool Move Collection Utility (EcDIMoveCollection.pl):** a perl utility, which allows DAAC operations staff to move a Data Pool collection from one file system to another.
12. **Data Pool Density Map Utility (EcDIDensityMapUtility.pl):** a perl utility, which calculates spatial density map information about Data Pool collections and stores this information in the Data Pool database. This utility normally runs as a cron job.
13. **Data Pool Statistics Table Population Utility (EcDIPopulateStatTables.pl):** a perl utility, which populates tables in the Data Pool database which maintain counts of granules by collection and collection group, for use by the Web Access drill down web pages. This utility normally runs as a cron job.
14. **Data Pool Hidden Scrambler Utility (EcDIHiddenScrambler.pl):** a perl utility, which creates new names for all hidden order-only directories, saves these names, renames the existing hidden directories, and updates existing FTP Pull links that point to the previous hidden directories to point to the corresponding renamed directory.
15. **Data Pool Database (DataPool[\_<MODE>]):** a Sybase database which stores Data Pool inventory and configuration information.

#### 4.14.1 Data Pool Subsystem Context

Figure 4.14-1 is the Data Pool Subsystem context diagram. The diagram shows the interaction of the Data Pool Subsystem with other EMD subsystems. Table 4.4-1 provides descriptions of the interface events shown in the Data Pool Subsystem context diagram.



**Figure 4.14-1. Data Pool Subsystem Context Diagram**

**Table 4.14-1. Data Pool Subsystem Interface Events (1 of 3)**

Interface Event	Interface Event Description
Monitor Data Pool Inserts	The operator uses the Data Pool Maintenance GUI to monitor the queue of Data Pool inserts and to monitor the active insert processes.
Configure Data Pool	The operator uses the Data Pool Maintenance GUI to set values of Data Pool configuration parameters, and to define Data Pool entities such as themes and compression algorithms.
Cleanup Data Pool	The operator runs the Data Pool Cleanup utility to clean expired granules out of the Data Pool, and to identify and cleanup granules, which are orphans (on Data Pool disk but not in the database) or phantoms (in the Data Pool database but not on disk).

**Table 4.14-1. Data Pool Subsystem Interface Events (2 of 3)**

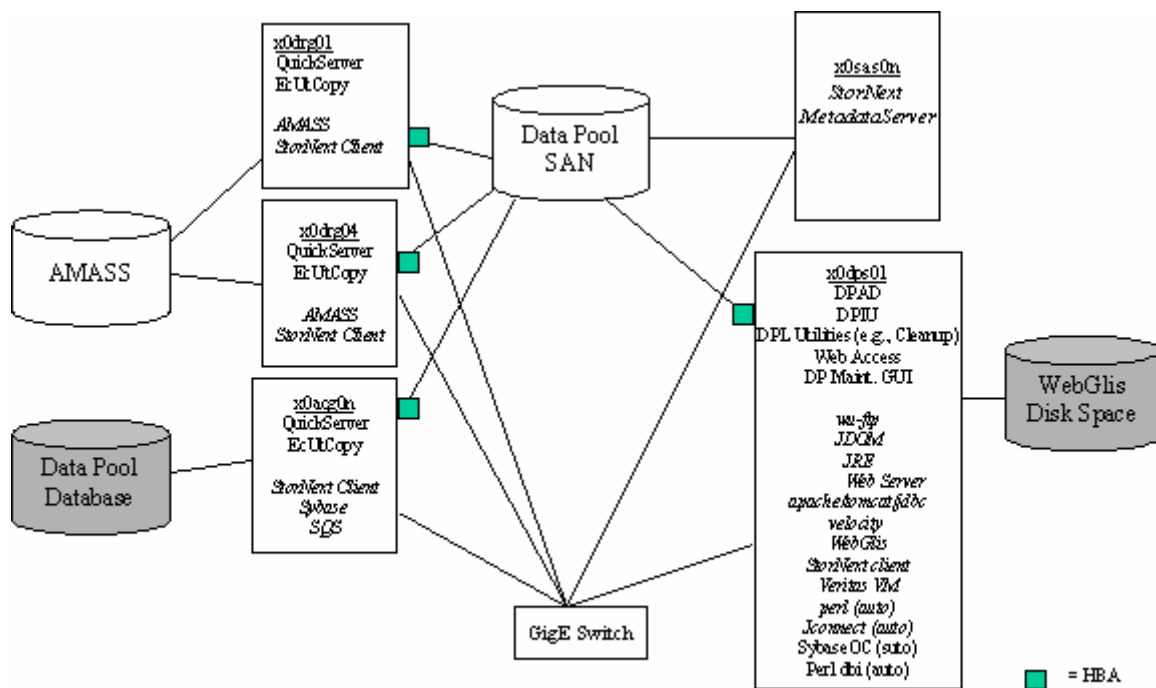
<b>Interface Event</b>	<b>Interface Event Description</b>
Manage Collections	The operator uses the Data Pool Maintenance GUI to add, remove, or change specifications for Data Pool collections. The operator uses the Remap Collection utility to map a collection from one collection group to another. The operator uses the Move Collection utility to move a collection from one file system to another.
Gather Access Statistics	The operator uses the access statistics rollup scripts for the firewall ftp and web access logs to gather statistics about end user access to data pool files, and to store those statistics in the Data Pool database.
Insert Action	The operator uses the Batch Insert utility to insert historical data from the EMD archive into the Data Pool, or to insert non-ECS data into the Data Pool.
Update Granule Expiration	The operator uses the Update Granule Expiration utility to update the expiration date or retention priority for a Data Pool granule or set of granules.
Update QA	The operator uses the QA Update utility to propagate updates of QA information from the SDSRV to the Data Pool.
Post Most Recent Inserts	The operator uses the Most Recent Inserts utility to post information about recent Data Pool Inserts to the Data Pool ftp directories.
FTP Data Pool data	The end user uses the customized WU-FTP service to download Data Pool data.
Drill Down	The end user uses the Web Access web pages to perform searches for Data Pool data.
Order Data Pool data	The end user uses the Web Access web pages to order Data Pool data for ftp or media distribution. The end user may choose to convert, reformat, or subset the data using the HDF-EOS to GeoTiff Conversion Tool (HEG).
Send email notification	The DPL subsystem (Web Access component) sends email to the end user indicating that the user's Data Pool order has been submitted. Email is sent by the WebAccess component only for downloads without HEG conversion, and only if the user requests email. (OMS sends order acknowledgement and distribution notice emails).
Insert action	The OMS subsystem inserts a Data Pool insert action into the Data Pool Insert Action Queue (DIInsertActionQueue) for granules to be staged to the Data Pool for EMD distribution requests in Synergy IV mode.
Insert Distribution Request	The DPL subsystem (WebAccess component) inserts distribution requests in the OMS database for Data Pool orders placed using the Data Pool Web Access web pages.
Check Order Status	The DPL subsystem (WebAccess component) checks status of orders in the MSS database.
Copy	The DPL subsystem copies data from AMASS to the Data Pool disk space.

**Table 4.14-1. Data Pool Subsystem Interface Events (3 of 3)**

Interface Event	Interface Event Description
Get Archive Location	The DPL subsystem looks up AMASS archive location information in the STMGT database, for granules which will be copied from AMASS to the Data Pool.
Get Metadata	The DPL subsystem gets metadata about EMD granules from the SDSRV database, and uses this metadata to store corresponding metadata in the Data Pool database and to create an xml metadata file on Data Pool disk.
Insert Action	The Spatial Subscription Server subsystem inserts Data Pool insert actions in the Data Pool Insert Action Queue (DIInsertActionQueue) for granules, which are being inserted into the EMD inventory for which a Data Pool insert subscription is placed.

#### 4.14.2 Data Pool Hardware Context

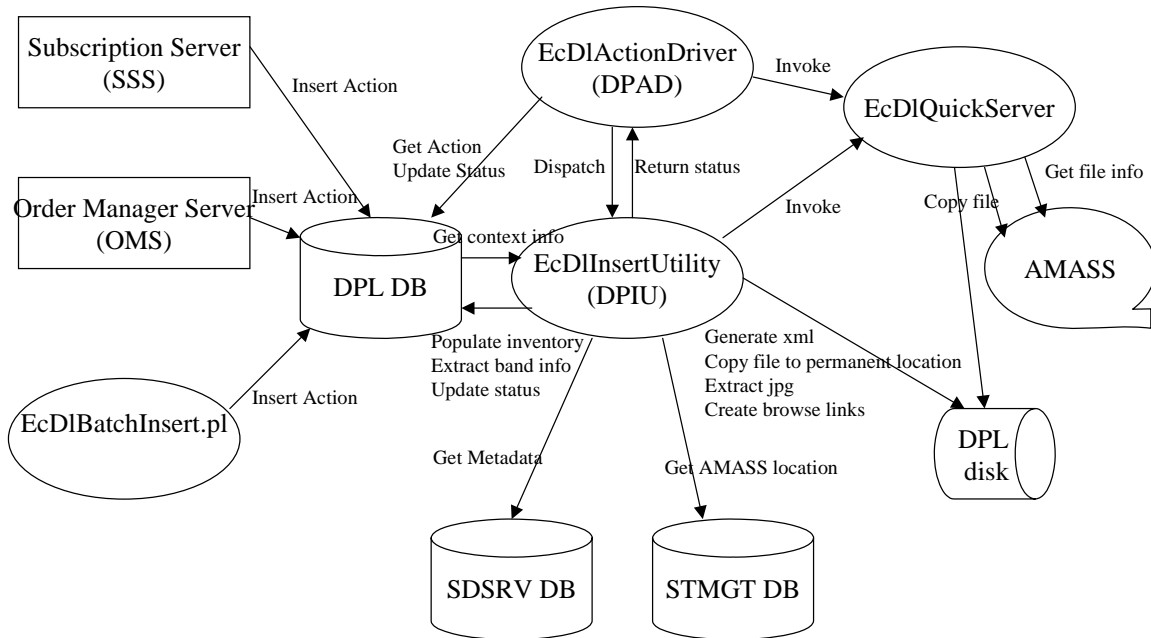
Figure 4.14-2 is the Data Pool hardware context diagram. The diagram shows the interaction of the Data Pool custom code and COTS (in *italics*) with EMD hardware components.



**Figure 4.14-2. Data Pool Hardware Context**

#### 4.14.3 Data Pool Insert CSCI Functional Overview

Figure 4.14-3 is the architecture diagram of the Data Pool insert process. Table 4.14-2 provides a process description for each of the major custom code components of the Data Pool insert process. Table 4.14-3 describes the interface events among the Data Pool insert process components.



**Figure 4.14-3. Data Pool Insert CSCI Architecture Diagram**

**Table 4.14-2. Data Pool Insert CSCI Process Description (1 of 2)**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDIActionDriver	Server	DPSHW	Developed	EcDIActionDriver is a C++ server that is responsible for dispatching Data Pool insert actions from the insert action queue in the Data Pool database (DIInsertActionQueue).



**Table 4.14-2. Data Pool Insert CSCI Process Description (2 of 2)**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDIInsertUtility	Java utility	DPSHW	Developed	EcDIInsertUtility is a java executable that is invoked by the EcDIActionDriver. One copy of the EcDIInsertUtility is invoked for each granule to be inserted in the Data Pool. The maximum number of EcDIInsertUtilities that may be executing at any one time is controlled by the NumOfAllowedCacheProcesses and NumOfAllowedNonCacheProcesses configuration items in DIConfig.
EcDIQuickServer	Server	DRPHW	Developed	The EcDIQuickServer is a C++ server which performs insert operations which are required on the AMASS hosts, such as AMASS API commands and copies.
EcUtCopyExec	Utility	DRPHW	Developed	The EcUtCopyExec is a C++ executable which performs copy operations on the AMASS hosts.
fileincache	Utility	DRPHW	Developed / COTS	fileincache is a utility developed using AMASS APIs which checks whether a file is in AMASS cache.
fileVolume	Utility	DRPHW	Developed / COTS	fileVolume is a utility developed using AMASS APIs which gets the tape label and file block number for a given file.
EcDIM2XT	Utility	DPSHW	Developed	Java utility that gets granule metadata from the SDSRV database and constructs an XML file.
hdf2jpeg	Utility	DPSHW	Developed	Java utility that extracts jpg's from an HDFEOS granule.
bandtool	Utility	DPSHW	Developed	C utility that extracts band information from an HDFEOS granule.

**Table 4.14-3. Data Pool Insert CSCI Process Interface Events (1 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Insert Action	One per granule inserted in SDSRV which qualifies for existing subscription with Data Pool Insert action	Database: DataPool	Trigger: TrigInsEcNbDpEventDetails.sql	When a granule is inserted into SDSRV which matches an existing subscription with Data Pool Insert action, the trigger inserts a row into the DIInsertActionQueue in the Data Pool database with actionSource = S.
Insert Action	One per granule in Syn IV order placed through Order Manager	Database: DataPool	Process: OmServer Stored Proc: OmInsDPLAction	When a granule is ordered in Syn IV mode via the Order Manager, OMS inserts a row into the DIInsertActionQueue in the Data Pool database, with actionSource = O.
Insert Action	One per granule in input file for the Batch Insert utility.	Database: DataPool	Utility: EcDIBatchInsert.pl	For each valid granule in its input file, the Batch Insert Utility inserts a row into the DIInsertActionQueue in the Data Pool database, with actionSource = B.
Get Action	Continuously, as long as there are actions in DIInsertActionQueue with status = null or status = RETRY. If no actions, once per configured time interval (IdleSleep in DIConfig)	Database: DataPool	Process: EcDIActionDriver	The Data Pool Action Driver gets batches of actions (with status = null or status = RETRY) from the DIInsertActionQueue.
Invoke	Once per insert action, if the inCacheFlag is 'N'	Process: EcDIQuickServer fileVolume	Process: EcDIActionDriver	The Data Pool Action Driver invokes the QuickServer with the fileVolume command to get the tape label and the file block number needed to determine which tape queue to use for the insert.

**Table 4.14-3. Data Pool Insert CSCI Process Interface Events (2 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Get file info	Once per insert action, if the inCacheFlag is 'N'	Storage Device: AMASS	Process: EcDIQuickServer fileVolume	The QuickServer uses the fileVolume utility to get the tape label and file block number required by the Data Pool Action Driver.
Dispatch	Once per action in DIInsertActionQueue	Process: EcDIInsertUtility	Process: EcDIActionDriver	The Data Pool Action Driver dispatches (via command line call) one Data Pool Insert Utility for each action in the DIInsertActionQueue with status = null or status = RETRY.
Get context info	Once per insert action	Database: DataPool	Process: EcDIInsertUtility Class: DIAbstractScienceGranule  Stored Proc: ProcGetCollInfo	The Data Pool Insert Utility gets context information for the insert, such as file system, compression algorithm, collection validity, from the Data Pool database.
Get metadata	Once per insert action, for EMD granules	Database: EcDsScienceData Server1	Process: EcDIInsertUtility Class: DIAbstractGranule	The Data Pool Insert Utility gets metadata about the EMD science granule such as file name and checksum value, from the SDSRV database.
Get AMASS location	Once per science file, for EMD granules	Database: stmgtdb1	Process: EcDIInsertUtility Class: DIAbstractGranule Stored Proc: ProcGetFileVolumeGroup	The Data Pool Insert Utility gets AMASS location of the granule from the STMGT database.
Invoke	Once per science file, for EMD granules	Process: EcDIQuickServer	Process: EcDIInsertUtility	The Data Pool Insert Utility invokes the Quick Server on the appropriate AMASS host to perform the fileincache and filevolume operations.

**Table 4.14-3. Data Pool Insert CSCI Process Interface Events (3 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Get file info	Once per science file per granule, for EMD granules	Storage Device: AMASS	Process: EcDIQuickServer fileincache filevolume	The Quick Server uses AMASS API utilities fileincache and filevolume to verify whether the file to be copied is in AMASS cache or on tape, and to get the volume information for the science file.
Invoke	Once per science file, for EMD granules	Process: EcDIQuickServer	Process: EcDIInsertUtility	The Data Pool Insert Utility invokes the Quick Server on the appropriate AMASS host to perform the AMASS copy operations.
Copy file	Once per science file per granule, for EMD granules	Storage Device: Data Pool disk (managed by COTS StorNext Storage Area Network)	Process: EcDIQuickServer EcUtCopyExec	The QuickServer uses the EcUtCopyExec to copy the science file from AMASS to the Data Pool file system for the collection.
Get metadata	Once per science granule, for EMD granules	Database: EcDsScienceDataServer1	Process: EcDIInsertUtility Class: DIEcsScienceGranule	The Data Pool Insert Utility gets spatial, temporal, and product specific metadata about the granule from the SDSRV database.
Generate xml	Once per granule	Storage Device: Data Pool disk	Process: EcDIInsertUtility EcDIM2XT Class: Metadata2XmlTool	The Data Pool Insert Utility invokes the Metadata to XML translator (M2XT) to generate an xml metatdata file, and writes the xml file to Data Pool disk.
Extract band info	Once per science granule, where convertEnabledFlag is on for the collection	Database: DataPool	Process: EcDIInsertUtility bandtool	The Data Pool Insert Utility uses the bandtool utility to extract band information from the science granule, and writes band information in the Data Pool database (DIGranuleHdf ObjectsXref)

**Table 4.14-3. Data Pool Insert CSCI Process Interface Events (4 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Copy file to permanent location	Once per science file per granule	Storage Device: Data Pool disk	Process: EcDIInsertUtility Class: DIUtils	The Data Pool Insert Utility copies each science file for the granule from the temporary directory on Data Pool disk to the permanent directory.
Populate inventory	Once per science granule	Database: DataPool	Process: EcDIInsertUtility Class: EcEcsScienceGranule	The Data Pool Insert Utility writes metadata about the science granule to the Data Pool database.
Get AMASS location	Once per associated browse file, for EMD granules	Database: stmgtdb1	Process: EcDIInsertUtility Class: DIAbstractGranule Stored Proc: ProcGetFileVolumeGroup	The Data Pool Insert Utility gets AMASS location of the associated browse files for the granule from the STMGT database.
Invoke	Once per insert for EMD granules	Process: EcDIQuickServer	Process: EcDIInsertUtility	The Data Pool Insert Utility invokes the Quick Server on the appropriate AMASS host to perform the AMASS copy operations for the browse files.
Get file info	Once per browse file per granule, for EMD granules	Storage Device: AMASS	Process: EcDIQuickServer fileincache filevolume	The Quick Server uses AMASS API utilities fileincache and filevolume to verify whether the browse file to be copied is in AMASS cache or on tape, and to get the volume location for the file.
Copy file	Once per browse file per granule, for EMD granules	Storage Device: Data Pool disk (managed by COTS StorNext Storage Area Network)	Process: EcDIQuickServer EcUtCopyExec	The QuickServer uses the EcUtCopyExec to copy the browse file from AMASS to the Data Pool file system for the collection.

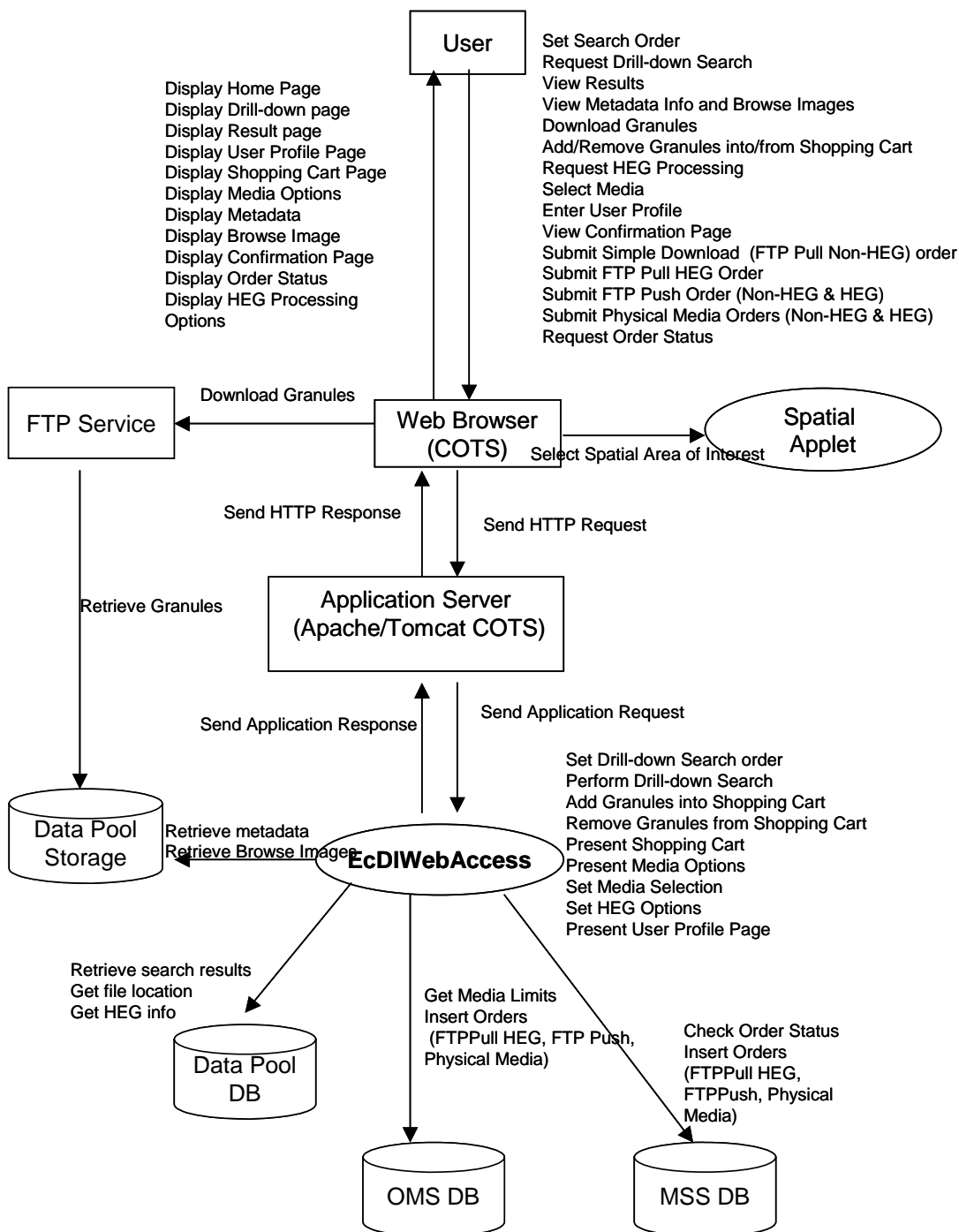
**Table 4.14-3. Data Pool Insert CSCI Process Interface Events (5 of 5)**

Event	Event Frequency	Interface	Initiated By	Event Description
Extract jpg	Once per browse image associated with a granule	Storage Device: Data Pool disk	Process: EcDIInsertUtility hdf2jpeg Class: DIUtils	The Data Pool Insert Utility uses the hdf2jpeg utility to extract jpg images from the hdfs browse files associated with the granule, and writes the jpg images to Data Pool disk.
Copy file to permanent location	Once per browse file per granule	Storage Device: Data Pool disk	Process: EcDIInsertUtility Class: DIUtils	The Data Pool Insert Utility copies each browse file for the granule from the temporary directory on Data Pool disk to the permanent directory.
Create browse links	Once per browse file per granule	Storage Device: Data Pool disk	Process: EcDIInsertUtility	The Data Pool Insert Utility creates links from the science file directory on Data Pool disk to the browse file directory.
Update status	Once per insert	Database: DataPool	Process: EcDIInsertUtility	The Data Pool Insert Utility updates the status, retryCount, and statusDetail fields in the DIInsertActionQueue.
Return status	Once per insert	Process: EcDIActionDriver	Process: EcDIInsertUtility	The Data Pool Insert Utility returns status of the insert to the Data Pool ActionDriver.
Update status	Once per insert, if Data Pool Insert Utility fails before updating status	Database: DataPool	Process: EcDIActionDriver	The Data Pool Action Driver updates the status and statusDetail fields in DIInsertActionQueue, if the Data Pool Insert Utility is unable to do so due to unexpected termination.

#### 4.14.4 WebAccess CSCI Functional Overview

Data Pool Web Access (EcDIWebAccess) is a java-based web application that runs with a web application server and related COTS. The Data Pool Web Access application interfaces with end-users, operators, and the Sybase server. It allows end-users to perform drill-down searches for Data Pool data, to view metadata and browse images online, to request conversion of Data Pool data and further to order them through ftp pull, ftp push and physical media.

Figure 4.14-4 is the WebAccess CSCI architecture diagram. The diagram shows the events sent to the WebAccess CSCI processes and the events the WebAccess CSCI processes send to other processes.



**Figure 4.14-4. WebAccess CSCI Architecture Diagram**

## WebAccess Process Descriptions

Table 4.14-4 provides descriptions of the processes shown in the WebAccess architecture diagram.

**Table 4.14-4. WebAccess CSCI Process Description**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcDIWebAccess	Web App	DPSHW	Developed	EcDIWebAccess is a web application running inside a web server. It provides user friendly web pages which allow users to search and retrieve data from Data Pool, view granules and related granule products once the list of granules has been retrieved, and order granules through ftp-pull, ftp-push and physical media. User may also request HEG conversions for granules in an order.
Application Server	Server	DPSHW	COTS	Application Server hosts the EcDIWebAccess web application.
Spatial Applet	Applet	n/a	Developed	The spatial applet allows users to interactively select desired area of interest on a map of the earth. The spatial applet uses WebGlis, a COTS product from USGS, to produce the map.
Web Browser	Browser	n/a	COTS	The Web Browser loads and displays EcDIWebAccess web pages.

## WebAccess Process Interface Descriptions

Table 4.14-5 describes the interface events among the WebAccess CSCI processes.



**Table 4.14-5. WebAccess CSCI Process Interface Events (1 of 10)**

<b>Event</b>	<b>Event Frequency</b>	<b>Interface</b>	<b>Initiated by</b>	<b>Event Description</b>
Set Search Order	One Per Search Sequence	Process: Web Browser (COTS)	User	The user configures the presentation of drill-down sequence following certain rules. The parameters are: Data Group, Theme, Data Set, Date, Time, Spatial, Cloud Cover, Day/Night Flag and Science QA.
Request Drill-down Search	One Per Request	Process: Web Browser (COTS)	User	The User specifies search criteria on each Drill-Down page (Theme/Group/ESDT, Temporal, TimeOfDay, AreaOfInterest, Cloud Cover etc.)
View Results	One Per Request	Process: Web Browser (COTS)	User	The User Clicks "Get the Result" link on the drill-down page, or the drill-down search attributes have been exhausted.
View Metadata Info and Browse Images	One Per Request	Process: Web Browser (COTS)	User	The User chooses to view the metadata information and/or browse image of a granule.
Download Granules	One Per Request	Process: Web Browser (COTS) Wu-FTP (COTS)	User	The User downloads the granules by initializing a ftp request.
Add/Remove Granules into/from Shopping Cart	One or Many Per Order Request	Process: Web Browser (COTS)	User	The User adds/removes granules into/from shopping cart.
Request HEG Processing	One or Many Per Order Request	Process: Web Browser (COTS)	User	The User selects format, projection, projection parameters, spatial subsetting or band subsetting for the granules in the shopping cart.

**Table 4.14-5. WebAccess CSCI Process Interface Events (2 of 10)**

<b>Event</b>	<b>Event Frequency</b>	<b>Interface</b>	<b>Initiated by</b>	<b>Event Description</b>
Select Media	One Per Order Request	Process: Web Browser (COTS)	User	The User selects one media option from the following: ftp-pull, ftp-push, CDROM and DVD.
Enter User Profile	One Per Order Request	Process: Web Browser (COTS)	User	The User enters user profile: name, email address, contact address, shipping address for a physical media order, ftp push related info for a ftp-push order.
View Confirmation Page	One Per Order Request	Process: Web Browser (COTS)	User	The User performs checkout and is presented with the detail of the orders.
Submit Simple Download (FTP-Pull Non-HEG) order	One Per Order Request	Process: Web Browser (COTS)	User	The User submits a simple download order that does not require HEG processing and is presented with data pool order id, download links.
Submit FTP Pull HEG Order	One Per Order Request	Process: Web Browser (COTS)	User	The User submits a ftp pull HEG order and is presented with an OMS order id and order status.
Submit FTP Push Order (nonHEG and HEG)	One Per Order Request	Process: Web Browser (COTS)	User	The User submits a ftppush order, either with or without HEG processing, and is presented with an OMS order id and order status.
Submit Physical Media Order (nonHEG and HEG)	One Per Order Request	Process: Web Browser (COTS)	User	The User submits a physical media order, either with or without HEG processing, and is presented with an OMS order id and order status.

**Table 4.14-5. WebAccess CSCI Process Interface Events (3 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Request Order Status	One or Many Per Request	Process: Web Browser (COTS)	User	The User requests the order status.
Send HTTP Request	One Per User Request	Process: Application Server (COTS)	Process: Web Browser (COTS)	The Web Browser sends HTTP request on behalf of the user to the Application Server.
Send Application Request	One Per User Request	Process: EcDIWebAccess	Process: Application Server (COTS)	The Application Server sends the request to EcDIWebAccess.
Set Drill-down Search Order	One Per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: SearchOrderServlet.java SearchOrderBean.java SearchOrderAction.java	The EcDIWebAccess sets the sequence of the searching parameters: Data Group, Theme, Data Set, Date, Time, Spatial, Cloud Cover, Day/Night Flag and Science QA.
Perform Drill-down Search	One Per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: DrilldownServlet.java SearchRequestBean.java AbstractDataBean.java ISummaryData.java	The EcDIWebAccess performs search based on the current drill-down searching parameters.

**Table 4.14-5. WebAccess CSCI Process Interface Events (4 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Select Spatial Area of Interest	One Per Request	Process: Web Browser (COTS)	Process: Web Browser (COTS) Spatial Applet Library: EcDISpatial.jar	The Web Browser hosts the Spatial Applet, handles user's interaction with a data coverage map of the earth and converts User's selection of spatial area of interest to HTTP request.
Retrieve Search Results	One or Many Per Order Request	Database: DataPool	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: GranuleRetrieverServlet.java GranuleDataBean.java	The EcDIWebAccess retrieves the search results from the Data Pool database, including notable data set level information such as average granule size, source parameter for cloud cover, product quality summary link or whether the data for a data set is typically compressed
Get File Location	One per file in result set	Database: DataPool	Process: EcDIWebAccess	EcDIWebAccess gets Data Pool disk location for metadata and browse files from the Data Pool database.
Retrieve Metadata	One Per Request	Storage Device: Data Pool disk	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: XMLServlet.java	The EcDIWebAccess retrieves the metadata of a granule from Data Pool disk.

**Table 4.14-5. WebAccess CSCI Process Interface Events (5 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Retrieve Browse Image	One Per Request	Storage Device: Data Pool disk	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: BrowseServlet.java	The EcDIWebAccess retrieves the browse image of a granule from Data Pool disk.
Add Granules into Shopping Cart	One or Many Per Order Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: GranuleRetrieverServlet.java CartBean.java	The EcDIWebAccess adds one or more granules into a shopping cart.
Remove Granules from Shopping Cart	One or Many Per Order Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java CartBean.java SetCartInfoAction.java EmptyCartAction.java	The EcDIWebAccess removes granule(s) from a shopping cart.
Present Shopping Cart	One or Many Per Order Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java CartBean.java DisplayCartAction.java	The EcDIWebAccess returns a shopping cart along with some data set level information and HEG processing options.

**Table 4.14-5. WebAccess CSCI Process Interface Events (6 of 10)**

<b>Event</b>	<b>Event Frequency</b>	<b>Interface</b>	<b>Initiated by</b>	<b>Event Description</b>
Get Media Limits	One Per Order Request	Database: Order Manager DB	Process: EcDIWebAccess	EcDIWebAccess gets media limit information from the OMS DB.
Present Media Options	One Set Per order request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java MediaAction.java	The EcDIWebAccess returns a media option list based on the configured media limits.
Set Media Selection	One Per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java MediaAction.java	The EcDIWebAccess saves the media selection for the order.
Get HEG info	One per Request, where convertEnabledFlag is set for one or more collections in cart	Database: DataPool	Process: EcDIWebAccess	EcDIWebAccess gets information from the Data Pool database to determine which, if any, HEG processing options to present.
Set HEG Options	One per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java	The EcDIWebAccess saves the HEG processing options for the order.

**Table 4.14-5. WebAccess CSCI Process Interface Events (7 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Present User Profile Page	One Per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java ProfileAction.java	The EcDIWebAccess returns a user profile page associated with the media via Application Server.
Process FTP Pull NON HEG (Simple Download) Order	One Per Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java SubmitOrderAction.java DownloadOrderImpl.java	The EcDIWebAccess saves the simple download order into DPL DB via Sybase Server and presents an order acknowledgement page for user to view the order it and download the data.
Insert Orders (FTPPull HEG, FTPPush, Physical Media)	One Per Order Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: CartServlet.java SetCartInfoAction.java OmOrderImpl.java OmHEGOrderImpl.java	The EcDIWebAccess saves the order into OM DB and MSS DB via Sybase Server and presents an order acknowledgement page for user to view the order it and order status.

**Table 4.14-5. WebAccess CSCI Process Interface Events (8 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Check Order Status	One or Many Per Order Request	Process: EcDIWebAccess	Application Server (COTS) Process: EcDIWebAccess Library: EcDIWaDrill.jar Class: OrderTrackingServlet.java OrderTrackingBean.java	The EcDIWebAccess tracks the order status in the MSS database, with the email address and order id.
Send Application Response	One Per User Request	Process: Application Server (COTS)	Process: EcDIWebAccess	EcDIWebAccess sends the response to the Application Server.
Send HTTP Response	One Per User Request	Process: Web Browser (COTS)	Process: Application Server (COTS)	The Application Server sends the response from EcDIWebAccess back to Web Browser.
Display Home Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the home page where user can set the drill-down order or start drill-down search with data set, data group or theme.
Display Drill-down Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the Drill-down page with the values of drill-down parameter.
Display Result Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the Result Page the captures notable data set level information such as average granule size, source parameter for cloud cover, product quality summary link or whether the data for a data set is typically compressed.



**Table 4.14-5. WebAccess CSCI Process Interface Events (9 of 10)**

<b>Event</b>	<b>Event Frequency</b>	<b>Interface</b>	<b>Initiated by</b>	<b>Event Description</b>
Display User Profile Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the user profile page during the final step in the order submission process.
Display Shopping Cart Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays a shopping cart along with some data set level information and HEG processing options.
Display Media Options	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the media options available for the current order.
Display Browse Image	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays browse image associated with a granule.
Display Metadata	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the full hierarchy metadata information of a granule.
Display Confirmation Page	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the confirmation page of the order.
Display Order Status	One Per Request	User	Process: Web Browser (COTS)	The Web Browser displays the order status upon submission of a ftp pull HEG order, ftp push (HEG & NON-HEG) and physical media order (HEG & NON-HEG)
Display HEG Processing Options	One Per Request	User	Process: Web Browser (COTS)	Besides the shopping cart page, the Web browser also displays HEG processing options in the input projection parameter page, band subsetting page and spatial subsetting page.

**Table 4.14-5. WebAccess CSCI Process Interface Events (10 of 10)**

Event	Event Frequency	Interface	Initiated by	Event Description
Download Granules	One Per User Request for Granule on Results Page	Process: FTP Service	User Process: Web Browser (COTS)	The user downloads granules from the results page using the Data Pool FTP Service.
Retrieve Granules	One Per User Request for Granule on Results Page	Storage Device: Data Pool Disk	Process: FTP Service	The Data Pool FTP Service retrieves the granule from Data Pool disk and downloads it to the user via ftp protocol.

#### 4.14.5 Data Stores

There are two data stores associated with the Data Pool subsystem. They are the Data Pool database (DPL DB) and the Order Manager database (OMS DB). Table 4.14-6 provides a description of these data stores.

**Table 4.14-6. Data Pool Data Stores**

Data Store	Type	Description
DPL DB	Sybase	The Data Pool (DPL) database implements the large majority of the persistent data requirements for the Data Pool subsystem. The Data Pool database contains: a) inventory data for the Data Pool granules, including data warehousing (Dimension and Fact) data which support Web Access drill down; b) configuration data for the Data Pool; c) interim processing data for the Data Pool utilities; d) data for monitoring Data Pool insert queues and processing; e) Data Pool access statistics; and f) information about data pool entities such as collection groups, collections, file systems, compression algorithms, and themes.
OMS DB	Sybase	The Order Manager (OMS) database stores persistent information about orders placed using the Data Pool WebAccess web pages.

## **4.15 Bulk Metadata Generation Tool Subsystem Overview**

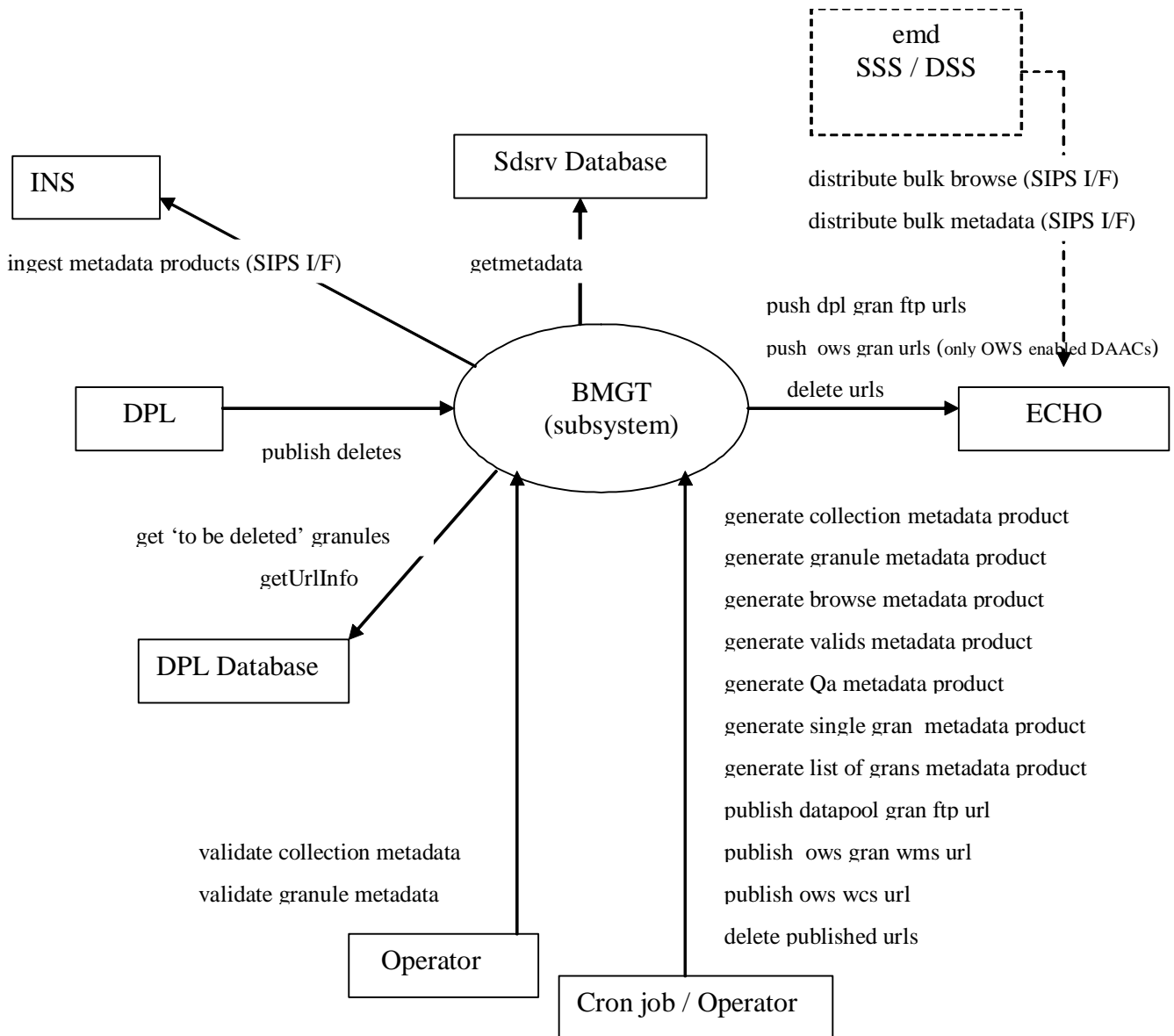
The EMD Bulk Metadata and Browse Export Capability were created to support the development of value-added providers (e.g., IIMS, ESIPs, RESACs, and InfoMarts). The Bulk Metadata Generation Tool (BMGT) provides interface to directly support the first part of the capability. The DSS subsystem addresses the second part of the capability. Currently, EOS Clearinghouse (ECHO) is the primary consumer of this capability.

The BMGT will facilitate EMD sites generate and export an external representation of their metadata holdings. The format used for the external representation of the metadata is XML. BMGT and Bulk Browse Generator Tools are run as daily cron jobs at each site to populate these data collections. One metadata product is created per ESDT group per day. Each product will contain an external representation of the metadata for each new, updated, or deleted granule that is a member of the ESDT group. One bulk browse product is produced per day that contains references to all new, updated, or deleted browse granules. Value-added providers may use any of the standard EMD search, order, and subscription capabilities to find and order these bulk metadata and browse products.

EMD distributes bulk metadata and browse products to ECHO via SIPS interface.

The BMGT, in addition to above capability, provides mechanism to tag granules that were exported to ECHO with Data Pool FTP, OGC conformant Web Mapping and Web Coverage Service URLs. These URLs are published to ECHO through its update interface.

Figure 4.15-1 is the BMGT Subsystem context diagram. The diagram shows the events generated between BMGT and other subsystems.



**Figure 4.15-1. BMGT Subsystem Context Diagram**

Table 4.15-1 provides descriptions of the interface events in the BMGT Subsystem context diagram.

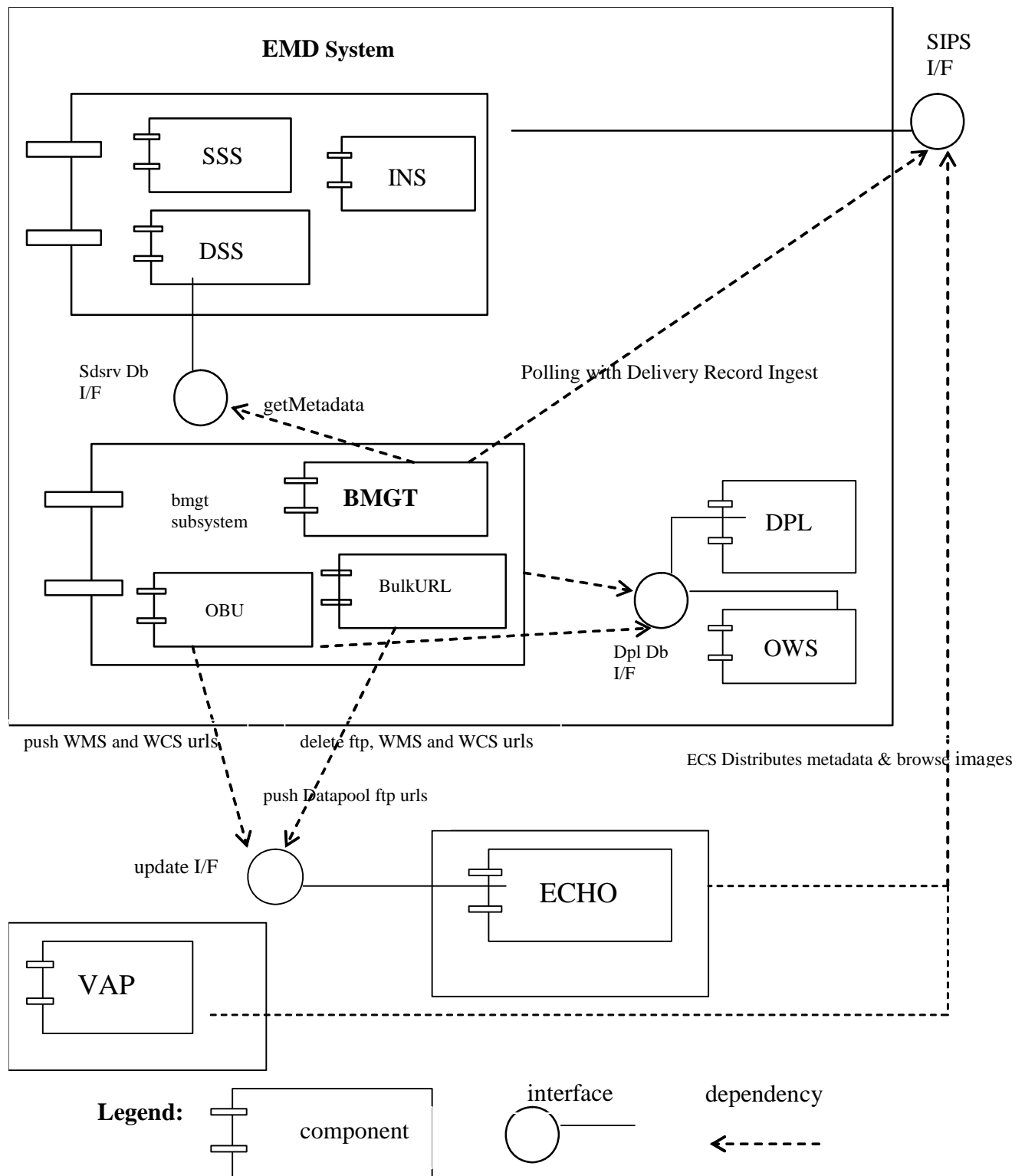
**Table 4.15-1. BMGT Subsystem Interface Events (1 of 2)**

Event	Interface Event Description
generate collection metadata product	BMGT generates ECSMETC metadata product for archive and distribution purposes. This product contains XML representation of the collection level metadata and the packaging options that may be used when ordering products from each collection. BMGT reads target ESDTs metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem). BMGT includes in the product WmsUrl and WcsUrl additional attributes if a collection qualifies as an OWS ESDT.
generate granule metadata product	BMGT generates ECSMETG metadata product for archive and distribution purposes. This product contains XML representation of the granule level metadata. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate browse metadata product	BMGT generates ECSBBR metadata product for archive and distribution purposes. This product contains XML representation of references to browse images. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate valids metadata product	BMGT generates ECSMETV metadata product for archive and distribution purposes. This product contains XML representation of EMD collection and granule valid values. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate Qa metadata product	BMGT generates ECSMETU metadata product for archive and distribution purposes. This product contains XML representation of Qa updates made to a granule. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate single gran metadata product	BMGT generates granule level metadata for the specified granule. A granule can be specified either by EMD dbID or Geoid. BMGT reads Metadata from SDSRV Database.
generate list of grans metadata product	BMGT generates granule level metadata for the specified granules. A list of EMD dbIDs or Geoids are made available to BMGT. BMGT reads Metadata from SDSRV Database.
publish Data Pool gran ftp url	BMGT/BulkURL generates a XML product containing Data Pool FTP URLs for the qualified granules. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows gran wms url	BMGT/OwsBindingUtility generates a XML product containing WmsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Mapping Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows wcs url	BMGT/OwsBindingUtility generates a XML product containing WcsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Coverage Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.

**Table 4.15-1. BMGT Subsystem Interface Events (2 of 2)**

<b>Event</b>	<b>Interface Event Description</b>
delete published urls	BMGT/BulkURL, upon invocation by Data Pool Cleanup utility, generates a XML product containing Data Pool granule URs that are targeted for removal from ECHO. The product is delivered to ECHO via FTP push or pull. ECHO will use this product to remove just the Data Pool related data like FTP, wms and wcs URLs. The product conforms to ECHO's update API.
validate collection metadata	BMGT validates its generated collection metadata (in xml format) with corresponding ODL Metadata file.
validate granule metadata	BMGT validates its generated granule metadata (in xml format) with corresponding ODL Metadata file.
GetUrlInfo	BMGT reads necessary info from Data Pool database to support 'publish' events mentioned above.
publish deletes	Data Pool Cleanup utility invokes BMGT/BulkURL to convey Data Pool granule deletes to ECHO. See 'delete published URLs' event above.
get 'to be deleted' granules	BMGT/BulkURL reads from Data Pool Database information pertaining to 'to be deleted granules'. Data Pool Cleanup utility persists this information before invoking BMGT/BulkURL.
ingest metadata products	BMGT generates ODL MET files and PDR files for all its Metadata products to facilitate 'Polling with Delivery Record Ingest'. This is part of SIPS interface.
Getmetadata	BMGT reads collection, granule, browse, valids and Qa related metadata from SDSRV Database.
distribute bulk browse	DSS distributes browse images to ECHO via SIPS interface. This is not BMGT's responsibility. Included here to show how this event is triggered if a ECSBBR product is acquired.
distribute bulk metadata	EMD distributes BMGT Metadata products to ECHO via SIPS interface.

#### **4.15.1 BMGT Architecture**



**Figure 4.15-2. BMGT Architecture Diagram**

Table 4.15-2 provides descriptions of processes shown in the architecture diagram.

**Table 4.15-2. BMGT Processes**

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcBmBMGT ("BMGT")	Application	ACMHW	Developed	The BMGT when run either as a cron job or manually, will generate ECSMETC, ECSMETG, ECSBBR, ECSMETV and the ECSMETU type xml products per its configuration. The products contain metadata about inventory that was inserted, updated, and/or deleted from the SDSRV database. Also, it generates a MET and PDR files for ingest purposes.
EcBmBulkURL ("BulkURL")	Application	DPSHW	Developed	The BulkURL exports to ECHO Data Pool FTP URLs and delete information about granule FTP, WCS and WMS URLs for the qualified granules. Export is achieved via FTP push or pull mechanism.
EcBmOwsBindingApp ("OBU")	Application	SANHW	Developed	The OBU exports to ECHO two pieces of metadata, WmsUrl and WcsUrl, which carry OWS service readiness information for individual granules that are newly enabled for OWS services (WMS and WCS, respectively).
EcBmBMGTValidateCollection	Script	ACMHW	Developed	A specified collection's generated metadata, in xml format, is validated. Script takes one ODL metadata file and one XML file as input parameters. The output of each script consists of a difference between the ODL metadata file and BMGT XML file.
EcBmBMGTValidateGranule	Script	ACMHW	Developed	A specified granule's generated metadata, in xml format, is validated. Script takes one ODL metadata file and one XML file as input parameters. The output of each script consists of a difference between the ODL metadata file and BMGT XML file.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE is where the SDSRV and Data Pool databases reside.

EBIS document 920-TDx-001 (HW Design Diagram) provides descriptions of the HWCI's and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.



## Use of COTS in the BMGT Subsystem

- JRE 1.4.x

The JRE constitutes Java virtual machine and the Java platform core libraries. It provides applications with Java platform. Included with it is JAXP (Java API for XML Processing).

- jConnect 5.5

The jConnect implements JDBC interface and provides application with drivers to access Sybase database SQL server.

- jdom b-8

jdom libraries allow java applications to create and edit xml documents.

- Sybase Server

The BMGT accesses SDSRV database to read inventory metadata and update Data Pool collection level attribute.

- JAF / Javamail

Java Activation Framework (JAF) and Javamail provide EcBmBulkURL and EcBmOwsBinding the capability to send email messages.

- jwsdp (w/JAXB)

EcBmOwsBinding utilizes Java Architecture for XML Binding (JAXB) functionality of jwsdp package. JAXB is a Java technology that allows easy binding of XML schemas to Java objects. Thus it helps application edit and create xml documents conforming to schemas.

- Sun One Web Server

BMGT (Document Type Definitions) DTD schemas are hosted on Sun One to provided access to consumers, like ECHO, who intend to validate the xml products.

- Perl

Validation tools – EcBmBMGTValidateCollection and EcBmBMGTValidateGranule need perl.

### 4.15.2 The BMGT Subsystem Software Description

As shown in Figure 4.15-2 architecture diagram, the BMGT Subsystem consists of three CSCIs: BMGT, BulkURL and the OBU. Essentially, these are Java applications

#### 4.15.2.1 BMGT CSCI Functional Overview

BMGT reads SDSRV Database to detect collections, granules and browse granules etc that have been inserted, updated or deleted during the specified period. Upon detection it will generate

XML representation of identified object's, say granule, metadata. The generated XML products enter into EMD as new granules. Products are archived for distribution purposes. XML products belong to one of ECSMETC, ECSMETG, ECSBBR, ECSMETV or ECSMETU collections.

BMGT product generation rules in terms of what type of products to generate, what target ESDTs to include, time range etc are configurable.

For each of the archivable xml products, BMGT creates a MET file. Also, it creates PDR file for each type of product. BMGT places XML products, MET and PDR files in a polling directory for Ingest pick up. Ingest interface is per SIPS interface. After successful archiving, the products are pushed to ECHO via SIPS interface. Note: DSS subsystem distributes Bulk Browse along with ECSBBR products.

Each of the product types i.e., BMGT specific ESDTs are described below:

- ECSMETC – Stores products that contain an XML representation of EMD collection level metadata and the packaging options that may be used when ordering products from each collection;
- ECSMETG – Stores products that contain an XML representation of EMD granule level metadata;
- ECSBBR - Stores products that contain an XML representation of references to browse images.
- ECSMETV – Stores products that contain an XML representation of EMD collection and granule valid values.
- ECSMETU – Stores products that contain an XML representation of EMD granule level QA Updates.

The ECSMETC and ECSMETG data collections store products that contain metadata for multiple collections and multiple granules. The metadata will be grouped by instrument and mission except for metadata related to the MODIS instrument, which is grouped, by mission and major discipline (ocean, atmosphere, land, and snow & ice). Each product in these collections has a group identifier Product Specific Attribute (PSA) called GroupId. The mapping of specific ESDTs to groups is provided as a configuration file with the BMGT.

The ECSBBR collection stores products that contain browse product references. The ESDT has a custom acquire service that will convert the browse product references into actual browse products during distribution.

The ECSMETV collection stores products that contain the entire set of valids contained within the SDSRV database for a particular instance in time.

Each EMD site will run BMGT as a daily cron job. It can also be run manually for any specified duration or to recover from cron job failures.

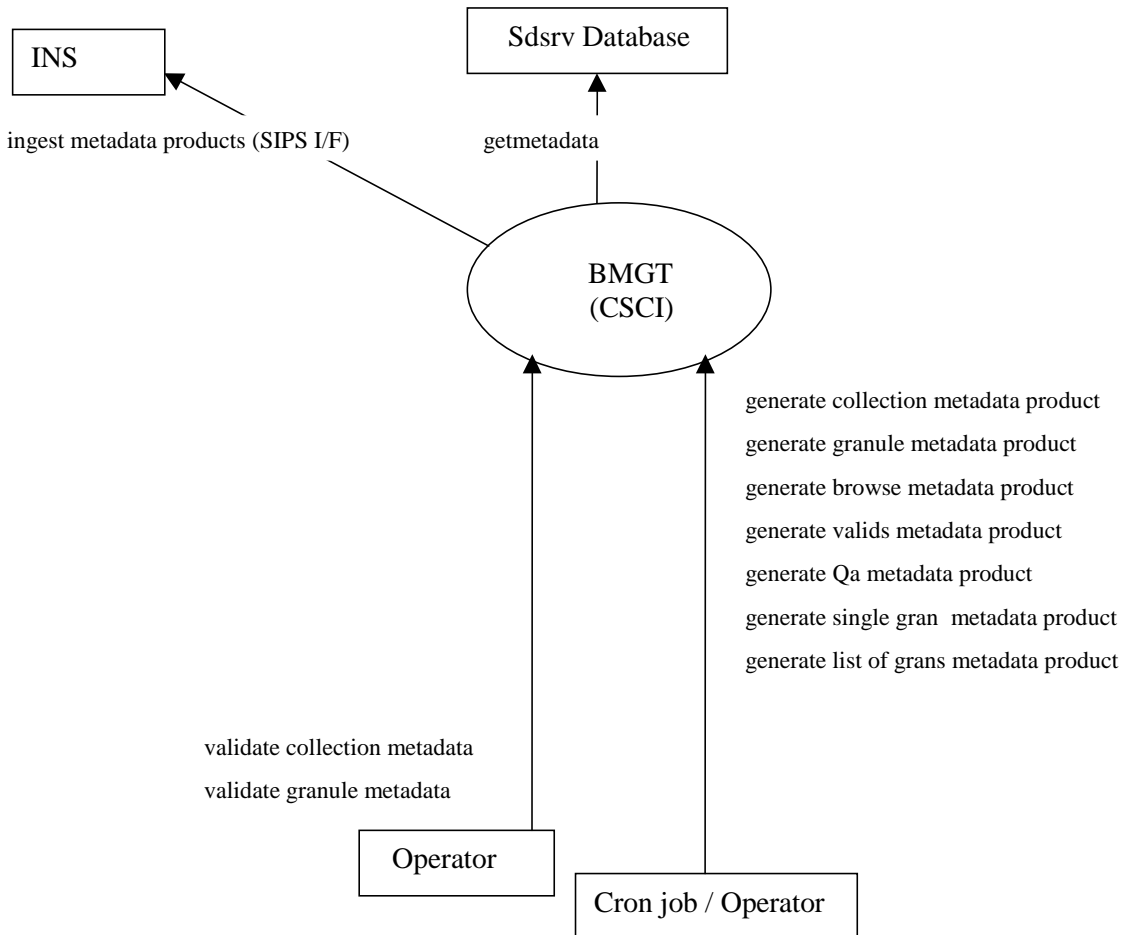
Refer EMD/SIPS ICD - 423-41-57, EMD/ECHO Metadata Inventory ICD (this still a work-in-progress artifact) and BMGT whitepaper - 170-WP-023-007 to better understand interfaces and OPS concept.

For reference, DTD schemas are included in Section 4.15.2.13.

#### 4.15.2.2 BMGT CSCI Context

Figure 4.15-3 is the BMGT CSCI context diagram.

Table 4.15-3 provides descriptions of the interface events shown in the BMGT CSCI context diagram.



**Figure 4.15-3. BMGT CSCI Context Diagram**

**Table 4.15-3. BMGT Subsystem Interface Events (1 of 2)**

Event	Interface Event Description
generate collection metadata product	BMGT generates ECSMETC metadata product for archive and distribution purposes. This product contains XML representation of the collection level metadata and the packaging options that may be used when ordering products from each collection. BMGT reads target ESDTs metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem). BMGT includes in the product WmsUrl and WcsUrl additional attributes if a collection qualifies as an OWS ESDT.
generate granule metadata product	BMGT generates ECSMETG metadata product for archive and distribution purposes. This product contains XML representation of the granule level metadata. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate browse metadata product	BMGT generates ECSBBR metadata product for archive and distribution purposes. This product contains XML representation of references to browse images. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate valids metadata product	BMGT generates ECSMETV metadata product for archive and distribution purposes. This product contains XML representation of EMD collection and granule valid values. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate Qa metadata product	BMGT generates ECSMETU metadata product for archive and distribution purposes. This product contains XML representation of Qa updates made to a granule. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate single gran metadata product	BMGT generates granule level metadata for the specified granule. A granule can be specified either by EMD dbID or Geoid. BMGT reads Metadata from SDSRV Database.
generate list of grans metadata product	BMGT generates granule level metadata for the specified granules. A list of EMD dbIDs or Geoids are made available to BMGT. BMGT reads Metadata from SDSRV Database.
publish Data Pool gran ftp url	BMGT/BulkURL generates a XML product containing Data Pool FTP URLs for the qualified granules. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows gran wms url	BMGT/OwsBindingUtility generates a XML product containing WmsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Mapping Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows wcs url	BMGT/OwsBindingUtility generates a XML product containing WcsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Coverage Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.

**Table 4.15-3. BMGT Subsystem Interface Events (2 of 2)**

Event	Interface Event Description
delete published urls	BMGT/BulkURL, upon invocation by Data Pool Cleanup utility, generates a XML product containing Data Pool granule URs that are targeted for removal from ECHO. The product is delivered to ECHO via FTP push or pull. ECHO will use this product to remove just the Data Pool related data like FTP, wms and wcs URLs. The product conforms to ECHO's update API.
validate collection metadata	BMGT validates its generated collection metadata (in xml format) with corresponding ODL Metadata file.
validate granule metadata	BMGT validates its generated granule metadata (in xml format) with corresponding ODL Metadata file.
GetUrlInfo	BMGT reads necessary info from Data Pool database to support 'publish' events mentioned above.
publish deletes	Data Pool Cleanup utility invokes BMGT/BulkURL to convey Data Pool granule deletes to ECHO. See 'delete published URLs' event above.
get 'to be deleted' granules	BMGT/BulkURL reads from Data Pool Database information pertaining to 'to be deleted granules'. Data Pool Cleanup utility persists this information before invoking BMGT/BulkURL.
ingest metadata products	BMGT generates ODL MET files and PDR files for all its Metadata products to facilitate 'Polling with Delivery Record Ingest'. This is part of SIPS interface.
Getmetadata	BMGT reads collection, granule, browse, valids and Qa related metadata from SDSRV Database.
distribute bulk browse	DSS distributes browse images to ECHO via SIPS interface. This is not BMGT's responsibility. Included here to show how this event is triggered if a ECSBBR product is acquired.
distribute bulk metadata	EMD distributes BMGT Metadata products to ECHO via SIPS interface.

#### **4.15.2.3 BMGT CSCI Process Interface Description**

Table 4.15-4 provides descriptions of the interface events shown in the BMGT context diagram.

**Table 4.15-4. BMGT CSCI Process Interface Events (1 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
generate collection metadata product	Once per service request	<i>Process:</i> EcBmBMGT  <i>Collaborator:</i> ProcBmgtGetEsdC ollMetadata	<i>Process:</i> Cron or operator	<p>Based on target ESDTs and period of interest specified in EcBmBMGTGroup.xml and EcBmBMGTUserParams.xml, BMGT generates ECSMETC xml product. Using a storedproc it reads metadata from SDSRV database. Inserts and updates are included in the product.</p> <p>The product conforms to BMGTCollectionMetadata.dtd schema (Section 4.15.2.13). One logical product is generated per group, which is a set of ESDTs. In addition, a MET file and PDR is generated.</p> <p>BMGT includes in the product WmsUrl and WcsUrl additional attributes if a collection qualifies as an OWS ESDT. OWS ESDTs are related to OWS themes created in Data Pool.</p>

**Table 4.15-4. BMGT CSCI Process Interface Events (2 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Generate granule metadata product	Once per service request	<i>Process:</i> EcBmBMGT  <i>Collaborator:</i> ProcBmgtGetEsdGranMetadata	<i>Process:</i> Cron or operator	<p>Based on target ESDTs and period of interest specified in EcBmBMGTGroup.xml and EcBmBMGTUserParams.xml, BMGT generates ECSMETG xml product. Using a storedproc it reads metadata from SDSRV database. Inserts, updates and deletes are included in the product.</p> <p>The product conforms to BMGTGranuleMetadata.dtd schema (Section 4.15.2.13). One logical product is generated per group, which is a set of ESDTs. In addition, a MET file and PDR is generated.</p>
Generate browse metadata product	Once per service request	<i>Process:</i> EcBmBMGT  <i>Collaborator:</i> ProcBmgtGetEsdBrowseMetadata	<i>Process:</i> Cron or operator	<p>For all browse images that were inserted or deleted within a specified time period, extract the browse identifiers and associated browse file names for each browse product and insert an XML file, called the Browse Reference File (BRF) file, as a product into the ECSBBR data collection.</p> <p>The product conforms to BMGTBrowseMetadata.dtd schema (Section 4.15.2.13). In addition, a MET file (per configured set of browse cross-references) and PDR is generated.</p>

**Table 4.15-4. BMGT CSCI Process Interface Events (3 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
generate valids metadata product	Once per service request	<i>Process:</i> EcBmBMGT  <i>Collaborator:</i> ProcBmgtGetVald Metadata	<i>Process:</i> Cron or operator	<p>In a process similar to granules, ECSMETV product is generated. Specifically, if any collections were inserted, updated, and/or deleted during the period then an XML representation of the valids information is generated, and inserted as a product, into the ECSMETV data collection setting the starting date and ending date of the insert, update, and/or delete activity covered by this file.</p> <p>The product conforms to BMGTValidsMetadata.dtd schema (Section 4.15.2.13). In addition, a MET file and PDR is generated.</p>
generate Qa metadata product	Once per service request	<i>Process:</i> EcBmBMGT  <i>Collaborator:</i> ProcBmgtGetEsdQaUpdates	<i>Process:</i> Cron or operator	<p>BMGT determines granules for which only QA flag metadata updates have been made and generates ECSMETU xml product for the identified granules. These products are used for notifying ECHO about just the QA updates. QAMUT persists QA update changes in SDSRV tables for BMGT consumption.</p> <p>The product conforms to BMGTUpdateMetadata.dtd schema (Section 4.15.2.13). In addition, a MET file and PDR is generated.</p>



**Table 4.15-4. BMGT CSCI Process Interface Events (4 of 4)**

Event	Event Frequency	Interface	Initiated By	Event Description
Validate collection metadata	One per service request	<i>Process:</i> EcBmBMGTValidateCollection	<i>Process:</i> operator	The script takes one ODL metadata file and one XML file as input parameters. The output of each script consists of a difference between the ODL metadata file and BMGT XML file. The BMGT XML collection file can contain multiple granules, but only the collection associated with the single ODL metadata file gets validated. All other collections are ignored. The same applies for granule XML files.  In order to obtain the ODL metadata file, the associated granule must be ordered from EMD.
Validate granule metadata	One per service request	<i>Process:</i> EcBmBMGTValidateGranule	<i>Process:</i> operator	Similar to above collection validation.

#### 4.15.2.4 Data Stores

BMGT uses SDSRV DB to generate its products. Table 4.15-5 describes the Data Stores.

**Table 4.15-5. CSCI Data Stores**

Data Store	Type	Description
SDSRV DB	Sybase	BMGT reads required metadata from SDSRV DB.

#### 4.15.2.5 BulkURL CSCI Functional Overview

The BulkURL utility (EcBmBulkURL) exports to ECHO the metadata content of products in the Data Pool, including their FTP URLs, based on an assumption that the product is an ECS data type and that the information about the product in the EMD Science Data Server has already been exported to ECHO by the Bulk Metadata Generation Tool (BMGT).

The utility can be run either with 'insert' or 'delete' option. With the Insert option, the BulkURL utility is run on a daily cron job or from the command line. Data Pool Cleanup utility invokes BulkURL with 'delete' option.

Specifically, it will export an xml representation of the FTP URL information for science files, metadata files, and browse files associated with Data Pool granules.

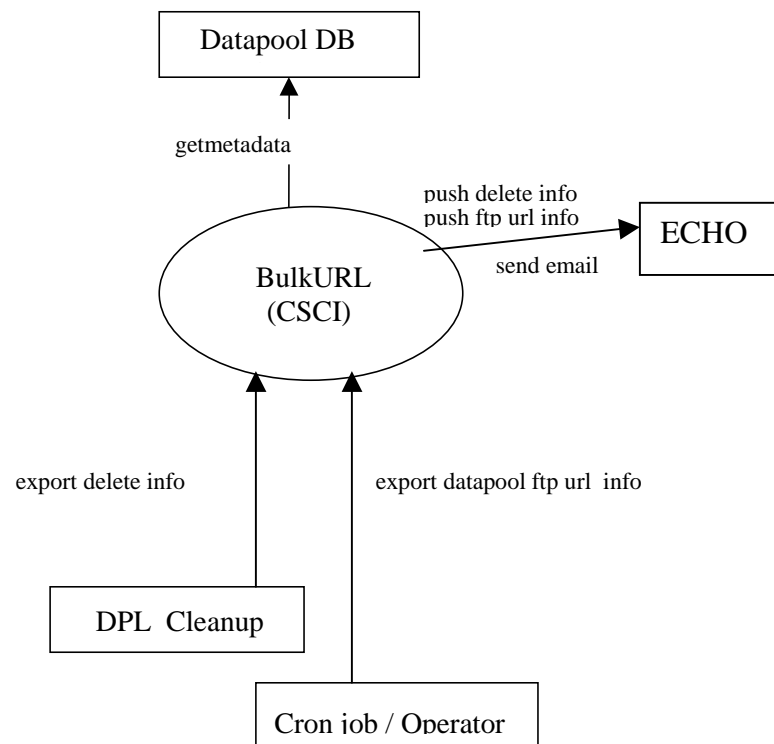
It also exports Data Pool delete information to ECHO. Delete message sent to ECHO is a request for it to remove any FTP, wms and wcs URLs associated with the deleted Data Pool granules.

BulkURL generated products conform to BMGTUpdateMetadata.dtd schema. Export is effected via FTP push or pull.

For reference, DTD schemas are included in Section 4.15.2.13.

#### 4.15.2.6 BulkURL CSCI Context

Figure 4.15-4 is the BulkURL CSCI context diagram.



**Figure 4.15-4. BulkURL CSCI Context Diagram**

**Table 4.15-6. BMGT Subsystem Interface Events (1 of 2)**

Event	Interface Event Description
generate collection metadata product	BMGT generates ECSMETC metadata product for archive and distribution purposes. This product contains XML representation of the collection level metadata and the packaging options that may be used when ordering products from each collection. BMGT reads target ESDTs metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem). BMGT includes in the product WmsUrl and WcsUrl additional attributes if a collection qualifies as an OWS ESDT.
generate granule metadata product	BMGT generates ECSMETG metadata product for archive and distribution purposes. This product contains XML representation of the granule level metadata. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate browse metadata product	BMGT generates ECSBBR metadata product for archive and distribution purposes. This product contains XML representation of references to browse images. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate valids metadata product	BMGT generates ECSMETV metadata product for archive and distribution purposes. This product contains XML representation of EMD collection and granule valid values. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate Qa metadata product	BMGT generates ECSMETU metadata product for archive and distribution purposes. This product contains XML representation of Qa updates made to a granule. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate single gran metadata product	BMGT generates granule level metadata for the specified granule. A granule can be specified either by EMD dbID or Geoid. BMGT reads Metadata from SDSRV Database.
generate list of grans metadata product	BMGT generates granule level metadata for the specified granules. A list of EMD dbIDs or Geoids are made available to BMGT. BMGT reads Metadata from SDSRV Database.
publish Data Pool gran ftp url	BMGT/BulkURL generates a XML product containing Data Pool FTP URLs for the qualified granules. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows gran wms url	BMGT/OwsBindingUtility generates a XML product containing WmsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Mapping Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows wcs url	BMGT/OwsBindingUtility generates a XML product containing WcsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Coverage Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.

**Table 4.15-6. BMGT Subsystem Interface Events (2 of 2)**

<b>Event</b>	<b>Interface Event Description</b>
delete published urls	BMGT/BulkURL, upon invocation by Data Pool Cleanup utility, generates a XML product containing Data Pool granule URs that are targeted for removal from ECHO. The product is delivered to ECHO via FTP push or pull. ECHO will use this product to remove just the Data Pool related data like FTP, wms and wcs URLs. The product conforms to ECHO's update API.
validate collection metadata	BMGT validates its generated collection metadata (in xml format) with corresponding ODL Metadata file.
validate granule metadata	BMGT validates its generated granule metadata (in xml format) with corresponding ODL Metadata file.
GetUrlInfo	BMGT reads necessary info from Data Pool database to support 'publish' events mentioned above.
publish deletes	Data Pool Cleanup utility invokes BMGT/BulkURL to convey Data Pool granule deletes to ECHO. See 'delete published URLs' event above.
get 'to be deleted' granules	BMGT/BulkURL reads from Data Pool Database information pertaining to 'to be deleted granules'. Data Pool Cleanup utility persists this information before invoking BMGT/BulkURL.
ingest metadata products	BMGT generates ODL MET files and PDR files for all its Metadata products to facilitate 'Polling with Delivery Record Ingest'. This is part of SIPS interface.
Getmetadata	BMGT reads collection, granule, browse, validates and Qa related metadata from SDSRV Database.
distribute bulk browse	DSS distributes browse images to ECHO via SIPS interface. This is not BMGT's responsibility. Included here to show how this event is triggered if a ECSBBR product is acquired.
distribute bulk metadata	EMD distributes BMGT Metadata products to ECHO via SIPS interface.

#### **4.15.2.7 BulkURL CSCI Process Interface Description**

Table 4.15-7 provides descriptions of the interface events shown in the BMGT context diagram.

**Table 4.15-7. BulkURL CSCI Process Interface Events (1 of 2)**

Event	Event Frequency	Interface	Initiated By	Event Description
export Data Pool ftp url info	Once per service request	<i>Process:</i> EcBmBulkURL  <i>Collaborator:</i> ProcOSGetDplURL	<i>Process:</i> Cron or operator	<p>The BulkURL utility executed with 'insert' option will export an xml representation of the FTP URL information for science files, metadata files, and browse files associated with Data Pool granules. After successful generation of xml product, the utility will push the product to ECHO via FTP. If it fails then the product will moved to a different folder and an email message is sent to ECHO for FTP pull.</p> <p>Only granules that meet a set of conditions are exported. The granule selection criteria is encapsulated in ProcOSGetDplURL stored proc.</p>

**Table 4.15-7. BulkURL CSCI Process Interface Events (2 of 2)**

Event	Event Frequency	Interface	Initiated By	Event Description
export delete info	Once per service request	<i>Process:</i> EcBmBulkURL  <i>Collaborator:</i> ProcOSGetDplDeletedURL, ProcBmGetOwsDplDeleted	<i>Process:</i> Cron or operator	<p>The Data Pool Cleanup Utility invokes BulkURL utility with 'delete' option. It will export an xml representation of granule UR and its FTP, wms and wcs URL information that needs to be removed from ECHO. After successful generation of xml product, the utility will push the product to ECHO via FTP. If it fails then the product will moved to a different folder and an email message is sent to ECHO for FTP pull.</p> <p>Only granules that meet a set of conditions are exported. The granule selection criteria is encapsulated in ProcOSGetDplDeletedURL, ProcBmGetOwsDplDeleted stored procs.</p> <p>BulkURL writes Delete information to a flat file to recover from a failure. It also persists deleted granule info' in a Data Pool table for DOWS Synchronizer's consumption.</p>

#### 4.15.2.8 Data Stores

BulkURL uses Data Pool DB to generate its products. Table 4.15-8 provides the Data Store description.

**Table 4.15-8. CSCI Data Stores**

Data Store	Type	Description
Data Pool DB	Sybase	BulkURL reads required metadata from Data Pool DB.

#### 4.15.2.9 OBU CSCI Functional Overview

The OWS Binding Utility (EcBmOwsBinding) exports to ECHO two pieces of metadata, WmsUrl and WcsUrl, which carry OWS service readiness information for individual granules that are newly enabled for OWS services (WMS and WCS, respectively). Before running the

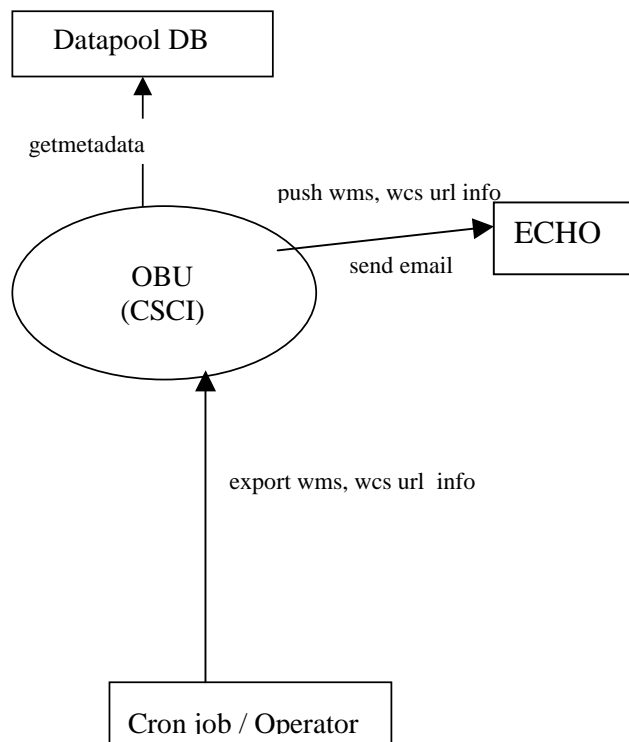
OWS Binding Utility, the collections to which any OWS enabled granules belong must be exported to ECHO with OWS attributes. Second, individual granule metadata for any OWS enabled granules must be exported to ECHO. Each of these tasks is performed by the Bulk Metadata Generation Tool (BMGT).

The utility is executed either manually or as a cron job. Its products conform to BMGTUpdateMetadata.dtd schema. Export is effected via FTP push or pull.

For reference puposes DTD schemas are included in Section 4.15.2.13.

#### 4.15.2.10 OBU CSCI Context

Figure 4.15-5 is the OBU CSCI context diagram.



**Figure 4.15-5. OBU CSCI Context Diagram**

**Table 4.15-9. BMGT Subsystem Interface Events (1 of 2)**

Event	Interface Event Description
generate collection metadata product	BMGT generates ECSMETC metadata product for archive and distribution purposes. This product contains XML representation of the collection level metadata and the packaging options that may be used when ordering products from each collection. BMGT reads target ESDTs metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem). BMGT includes in the product WmsUrl and WcsUrl additional attributes if a collection qualifies as an OWS ESDT.
generate granule metadata product	BMGT generates ECSMETG metadata product for archive and distribution purposes. This product contains XML representation of the granule level metadata. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate browse metadata product	BMGT generates ECSBBR metadata product for archive and distribution purposes. This product contains XML representation of references to browse images. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate valids metadata product	BMGT generates ECSMETV metadata product for archive and distribution purposes. This product contains XML representation of EMD collection and granule valid values. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate Qa metadata product	BMGT generates ECSMETU metadata product for archive and distribution purposes. This product contains XML representation of Qa updates made to a granule. BMGT reads Metadata from SDSRV Database. MET files and a single PDR file are generated for EMD ingest purposes (INS Subsystem).
generate single gran metadata product	BMGT generates granule level metadata for the specified granule. A granule can be specified either by EMD dbID or Geoid. BMGT reads Metadata from SDSRV Database.
generate list of grans metadata product	BMGT generates granule level metadata for the specified granules. A list of EMD dbIDs or Geoids are made available to BMGT. BMGT reads Metadata from SDSRV Database.
publish Data Pool gran ftp url	BMGT/BulkURL generates a XML product containing Data Pool FTP URLs for the qualified granules. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows gran wms url	BMGT/OwsBindingUtility generates a XML product containing WmsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Mapping Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.
publish ows wcs url	BMGT/OwsBindingUtility generates a XML product containing WcsUrl PSA for the qualified granules. This PSA encapsulates binding information about EMD Web Coverage Service. The product is delivered to ECHO via FTP push or pull. The product conforms to ECHO's update API. Information required to formulate FTP URL is read from Data Pool Database.



**Table 4.15-9. BMGT Subsystem Interface Events (2 of 2)**

<b>Event</b>	<b>Interface Event Description</b>
delete published urls	BMGT/BulkURL, upon invocation by Data Pool Cleanup utility, generates a XML product containing Data Pool granule URs that are targeted for removal from ECHO. The product is delivered to ECHO via FTP push or pull. ECHO will use this product to remove just the Data Pool related data like FTP, wms and wcs URLs. The product conforms to ECHO's update API.
validate collection metadata	BMGT validates its generated collection metadata (in xml format) with corresponding ODL Metadata file.
validate granule metadata	BMGT validates its generated granule metadata (in xml format) with corresponding ODL Metadata file.
GetUrlInfo	BMGT reads necessary info from Data Pool database to support 'publish' events mentioned above.
publish deletes	Data Pool Cleanup utility invokes BMGT/BulkURL to convey Data Pool granule deletes to ECHO. See 'delete published URLs' event above.
get 'to be deleted' granules	BMGT/BulkURL reads from Data Pool Database information pertaining to 'to be deleted granules'. Data Pool Cleanup utility persists this information before invoking BMGT/BulkURL.
ingest metadata products	BMGT generates ODL MET files and PDR files for all its Metadata products to facilitate 'Polling with Delivery Record Ingest'. This is part of SIPS interface.
Getmetadata	BMGT reads collection, granule, browse, valids and Qa related metadata from SDSRV Database.
distribute bulk browse	DSS distributes browse images to ECHO via SIPS interface. This is not BMGT's responsibility. Included here to show how this event is triggered if a ECSBBR product is acquired.
distribute bulk metadata	EMD distributes BMGT Metadata products to ECHO via SIPS interface.

#### **4.15.2.11 OBU CSCI Process Interface Description**

Table 4.15-9 provides descriptions of the interface events shown in the OBU context diagram.

**Table 4.15-10. OBU CSCI Process Interface Events**

Event	Event Frequency	Interface	Initiated By	Event Description
export wms, wcs url info	Once per service request	<i>Process:</i> EcBmOwsBinding  <i>Collaborator:</i> ProcBmGetOwsIn dexedGranules, ProcBmOwsMarkP ublished	<i>Process:</i> Cron or operator	<p>The OBU will export an xml representation of the wms and wcs URLs for the identified OWS granules. Essentially, these granules have to be in 'indexed' state in DIGranuleOWS. After successful generation of xml product, the utility will push the product to ECHO via FTP. If it fails then the product will moved to a different folder and an email message is sent to ECHO for FTP pull.</p> <p>After successful export, granules are promoted to published state in DIGranuleOWS.</p> <p>Only granules that meet a set of conditions are exported. The granule selection criteria is encapsulated in ProcOSGetDplURL stored proc.</p>

#### 4.15.2.12 Data Stores

OBU uses Data Pool DB to generate its products.

**Table 4.15-11. CSCI Data Stores**

Data Store	Type	Description
Data Pool DB	Sybase	OBU reads required metadata from Data Pool DB.

#### 4.15.2.13 BMGT DTD Schemas

##### 4.15.2.13.1 BMGTCollectionMetadata.dtd

```
<!ELEMENT CollectionMetaFile (DTDVersion, DataCenterId, TemporalCoverage, DefaultPackage, CollectionMetaFile*)>
```

```
<!-- Version identifier of the DTD used to generate the file -->
```

```
<!ELEMENT DTDVersion (#PCDATA)>
```

```

<!-- DataCenterId of the site that stores this metadata (e.g., LP DAAC) -->
<!ELEMENT DataCenterId (#PCDATA)>

<!-- the start and end dates of this MetaDataFile (YYYY-MM-DD) -->
<!ELEMENT TemporalCoverage (StartDate, EndDate)>
<!ELEMENT StartDate (#PCDATA)>
<!ELEMENT EndDate (#PCDATA)>

<!-- Default Packaging Information will apply to every data collection unless
over written in the collection-level metadata -->
<!ELEMENT DefaultPackage (MediaTypes+, ProductionOptions, EstimatedCost?)>
<!ELEMENT EstimatedCost (#PCDATA)>

<!ELEMENT MediaTypes (MediaType, MediaFormats+)>
<!ELEMENT MediaType (#PCDATA)>

<!ELEMENT MediaFormats (MediaFormat, MediaParameters*)>
<!ELEMENT MediaFormat (#PCDATA)>

<!ELEMENT MediaParameters (ParameterName?, Specialized?, Obscured?, Type?,
Mandatory?, MaxLen?, Label?, MediaValid?)>
<!ELEMENT ParameterName (#PCDATA)>
<!ELEMENT Specialized (#PCDATA)>
<!ELEMENT Obscured (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Mandatory (#PCDATA)>
<!ELEMENT MaxLen (#PCDATA)>
<!ELEMENT Label (#PCDATA)>

<!ELEMENT MediaValid (MediaValid)+>
<!ELEMENT MediaValid (#PCDATA)>

<!ELEMENT ProductionOptions (ProductionHistoryOptionName?,
AncillaryDataOptionName?, NativeGranuleOptionName?)>
<!ELEMENT ProductionHistoryOptionName (#PCDATA)>
<!ELEMENT AncillaryDataOptionName (#PCDATA)>
<!ELEMENT NativeGranuleOptionName (#PCDATA)>

<!ELEMENT CollectionMetaData (ShortName, VersionID, InsertTime, LastUpdate?,
LongName, CollectionDescription, RevisionDate?, SuggestedUsagel?,
SuggestedUsage2?, ProcessingCenter?, ProcessingLevelId?,
ProcessingLevelDescription?, ArchiveCenter, VersionDescription,
CitationforExternalPublication?, CollectionState?,
MaintenanceandUpdateFrequency?, AccessConstraints?, CollectionPackage?,
Spatial?, Temporal?, Contact*, DisciplineTopicParameters*, Platform*,
StorageMedium*, AdditionalAttributes*, BrowseProduct?, SpatialKeyword*,
TemporalKeyword*, CSDDTDescription*, Locality*, CollReview*, Documents?,
CollectionAssociation*, AnalysisSource*, Campaign*)>
<!ELEMENT ShortName (#PCDATA)>
<!ELEMENT VersionID (#PCDATA)>
<!ELEMENT InsertTime (#PCDATA)>
<!ELEMENT LastUpdate (#PCDATA)>
<!ELEMENT LongName (#PCDATA)>
<!ELEMENT CollectionDescription (#PCDATA)>
<!ELEMENT RevisionDate (#PCDATA)>
<!ELEMENT SuggestedUsagel (#PCDATA)>
<!ELEMENT SuggestedUsage2 (#PCDATA)>
<!ELEMENT ProcessingCenter (#PCDATA)>

```

```

<!ELEMENT ProcessingLevelId (#PCDATA)>
<!ELEMENT ProcessingLevelDescription (#PCDATA)>
<!ELEMENT ArchiveCenter (#PCDATA)>
<!ELEMENT VersionDescription (#PCDATA)>
<!ELEMENT CitationforExternalPublication (#PCDATA)>
<!ELEMENT CollectionState (#PCDATA)>
<!ELEMENT MaintenanceandUpdateFrequency (#PCDATA)>
<!ELEMENT AccessConstraints (#PCDATA)>
<!ELEMENT StorageMedium (#PCDATA)>
<!ELEMENT SpatialKeyword (#PCDATA)>
<!ELEMENT TemporalKeyword (#PCDATA)>

<!ELEMENT CollectionPackage (MediaTypes+, ProductionOptions, EstimatedCost?)>

<!ELEMENT Spatial (SpatialCoverageType, HorizontalSpatialDomain?,
VerticalSpatialDomain*, CoordinateSystemContainer?, OrbitParameters?,
GranuleSpatialRepresentation?)>
<!ELEMENT SpatialCoverageType (#PCDATA)>

<!ELEMENT HorizontalSpatialDomain ((ZoneIdentifier?,Geometry) | Global)>
<!ELEMENT ZoneIdentifier (#PCDATA)>
<!ELEMENT Geometry (CoordinateSystem?, (Point | Circle | BoundingRectangle |
GPolygon))>

<!ELEMENT CoordinateSystem (Geodetic | Cartesian)>
<!ELEMENT Geodetic EMPTY>
<!ELEMENT Cartesian EMPTY>
<!ELEMENT Global EMPTY>

<!ELEMENT Point (PointLongitude, PointLatitude)>
<!ELEMENT PointLongitude (#PCDATA)>
<!ELEMENT PointLatitude (#PCDATA)>

<!ELEMENT Circle (CenterLatitude, CenterLongitude, Radius, RadiusUnits)>
<!ELEMENT CenterLatitude (#PCDATA)>
<!ELEMENT CenterLongitude (#PCDATA)>
<!ELEMENT Radius (#PCDATA)>
<!ELEMENT RadiusUnits (#PCDATA)>

<!ELEMENT BoundingRectangle (WestBoundingCoordinate, NorthBoundingCoordinate,
EastBoundingCoordinate, SouthBoundingCoordinate)>
<!ELEMENT WestBoundingCoordinate (#PCDATA)>
<!ELEMENT NorthBoundingCoordinate (#PCDATA)>
<!ELEMENT EastBoundingCoordinate (#PCDATA)>
<!ELEMENT SouthBoundingCoordinate (#PCDATA)>

<!ELEMENT GPolygon (Boundary+)>
<!ELEMENT Boundary (Point, Point, Point, Point*)>

<!ELEMENT VerticalSpatialDomain (VerticalSpatialDomainType,
VerticalSpatialDomainValue)>
<!ELEMENT VerticalSpatialDomainType (#PCDATA)>
<!ELEMENT VerticalSpatialDomainValue (#PCDATA)>

<!ELEMENT CoordinateSystemContainer (VerticalCoordinateSystemContainer?,
HorizontalCoordinateSystemContainer?)>

```

```

<!ELEMENT VerticalCoordinateSystemContainer (AltitudeSystemDefinition?,
DepthSystemDefinition?)>

<!ELEMENT AltitudeSystemDefinition (AltitudeDatumName, AltitudeDistanceUnits,
AltitudeEncodingMethod, AltitudeResolution)>
<!ELEMENT AltitudeDatumName (#PCDATA)>
<!ELEMENT AltitudeDistanceUnits (#PCDATA)>
<!ELEMENT AltitudeEncodingMethod (#PCDATA)>
<!ELEMENT AltitudeResolution (#PCDATA)>

<!ELEMENT DepthSystemDefinition (DepthDatumName, DepthDistanceUnits,
DepthEncodingMethod, DepthResolution)>
<!ELEMENT DepthDatumName (#PCDATA)>
<!ELEMENT DepthDistanceUnits (#PCDATA)>
<!ELEMENT DepthEncodingMethod (#PCDATA)>
<!ELEMENT DepthResolution (#PCDATA)>

<!ELEMENT HorizontalCoordinateSystemContainer (GeodeticModel?,
(GeographicCoordinateSystem | PlanarCoordinateSystems |
LocalCoordinateSystem))>

<!ELEMENT GeodeticModel (HorizontalDatumName?, EllipsoidName, SemiMajorAxis,
DenominatorofFlatteningRatio)>
<!ELEMENT HorizontalDatumName (#PCDATA)>
<!ELEMENT EllipsoidName (#PCDATA)>
<!ELEMENT SemiMajorAxis (#PCDATA)>
<!ELEMENT DenominatorofFlatteningRatio (#PCDATA)>

<!ELEMENT GeographicCoordinateSystem (LatitudeResolution, LongitudeResolution,
GeographicCoordinateUnits)>
<!ELEMENT LatitudeResolution (#PCDATA)>
<!ELEMENT LongitudeResolution (#PCDATA)>
<!ELEMENT GeographicCoordinateUnits (#PCDATA)>

<!ELEMENT PlanarCoordinateSystems (PlanarCoordinateSystem)>

<!ELEMENT PlanarCoordinateSystem (PlanarCoordinateSystemContainer+)>

<!ELEMENT PlanarCoordinateSystemContainer (PlanarCoordinateInformation,
(MapProjection | LocalPlanarCoordinateSystem | GridCoordinateSystem))>

<!ELEMENT PlanarCoordinateInformation (PlanarDistanceUnits,
PlanarCoordinateEncodingMethod, (DistanceandBearingRepresentation |
CoordinateRepresentation))>
<!ELEMENT PlanarDistanceUnits (#PCDATA)>
<!ELEMENT PlanarCoordinateEncodingMethod (#PCDATA)>

<!ELEMENT DistanceandBearingRepresentation (DistanceResolution,
BearingResolution, BearingUnits, BearingReferenceDirection,
BearingReferenceMeridian)>
<!ELEMENT DistanceResolution (#PCDATA)>
<!ELEMENT BearingResolution (#PCDATA)>
<!ELEMENT BearingUnits (#PCDATA)>
<!ELEMENT BearingReferenceDirection (#PCDATA)>
<!ELEMENT BearingReferenceMeridian (#PCDATA)>

<!ELEMENT CoordinateRepresentation (AbscissaResolution, OrdinateResolution)>
<!ELEMENT AbscissaResolution (#PCDATA)>

```

```

<!ELEMENT OrdinateResolution (#PCDATA)>

<!ELEMENT MapProjection (MapProjectionName, MapProjectionPointer?)>
<!ELEMENT MapProjectionName (#PCDATA)>
<!ELEMENT MapProjectionPointer (#PCDATA)>

<!ELEMENT LocalPlanarCoordinateSystem (LocalPlanarCoordinateSystemDescription,
LocalPlanarGeoreferenceInformation)>
<!ELEMENT LocalPlanarCoordinateSystemDescription (#PCDATA)>
<!ELEMENT LocalPlanarGeoreferenceInformation (#PCDATA)>

<!ELEMENT GridCoordinateSystem (GridCoordinateSystemName)>
<!ELEMENT GridCoordinateSystemName (#PCDATA)>

<!ELEMENT LocalCoordinateSystem (LocalCoordinateSystemDescription,
LocalGeoreferenceInformation)>
<!ELEMENT LocalCoordinateSystemDescription (#PCDATA)>
<!ELEMENT LocalGeoreferenceInformation (#PCDATA)>

<!ELEMENT OrbitParameters (SwathWidth, Period, InclinationAngle)>
<!ELEMENT SwathWidth (#PCDATA)>
<!ELEMENT Period (#PCDATA)>
<!ELEMENT InclinationAngle (#PCDATA)>

<!ELEMENT GranuleSpatialRepresentation (Cartesian | Geodetic | Orbit |
NoSpatial)>
<!ELEMENT Orbit EMPTY>
<!ELEMENT NoSpatial EMPTY>

<!ELEMENT Temporal (TimeType, DateType, TemporalRangeType, PrecisionofSeconds,
EndsatPresentFlag, (RangeDateTime | SingleDateTime+))>
<!ELEMENT TimeType (#PCDATA)>
<!ELEMENT DateType (#PCDATA)>
<!ELEMENT TemporalRangeType (#PCDATA)>
<!ELEMENT PrecisionofSeconds (#PCDATA)>
<!ELEMENT EndsatPresentFlag (#PCDATA)>

<!ELEMENT RangeDateTime (RangeBeginningDate, RangeBeginningTime,
RangeEndingDate?, RangeEndingTime?)>
<!ELEMENT RangeBeginningDate (#PCDATA)>
<!ELEMENT RangeBeginningTime (#PCDATA)>
<!ELEMENT RangeEndingDate (#PCDATA)>
<!ELEMENT RangeEndingTime (#PCDATA)>

<!ELEMENT SingleDateTime (CalendarDate, Timeofday)>
<!ELEMENT CalendarDate (#PCDATA)>
<!ELEMENT Timeofday (#PCDATA)>

<!ELEMENT Contact (ContactRole, HoursOfService?, ContactInstructions?,
(Organization | ContactPerson), Address?, Email?, Telephone?, Fax?,
ContactURL?)>

<!ELEMENT ContactRole (#PCDATA)>
<!ELEMENT HoursOfService (#PCDATA)>
<!ELEMENT ContactInstructions (#PCDATA)>

<!ELEMENT Organization (OrganizationName)>
<!ELEMENT OrganizationName (#PCDATA)>

```

```

<!ELEMENT ContactPerson (FirstName, MiddleName?, LastName, JobPosition?)>
<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT MiddleName (#PCDATA)>
<!ELEMENT LastName (#PCDATA)>
<!ELEMENT JobPosition (#PCDATA)>

<!ELEMENT Address (StreetAddress, City, StateProvince, PostalCode, Country)>
<!ELEMENT StreetAddress (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT StateProvince (#PCDATA)>
<!ELEMENT PostalCode (#PCDATA)>
<!ELEMENT Country (#PCDATA)>

<!ELEMENT Email (#PCDATA)>
<!ELEMENT Telephone (#PCDATA)>
<!ELEMENT Fax (#PCDATA)>
<!ELEMENT ContactURL (#PCDATA)>

<!ELEMENT DisciplineTopicParameters (DisciplineKeyword, TopicKeyword,
TermKeyword, VariableKeyword?, ECSPParameterKeyword*)>
<!ELEMENT DisciplineKeyword (#PCDATA)>
<!ELEMENT TopicKeyword (#PCDATA)>
<!ELEMENT TermKeyword (#PCDATA)>
<!ELEMENT VariableKeyword (#PCDATA)>
<!ELEMENT ECSPParameterKeyword (#PCDATA)>

<!ELEMENT Platform (PlatformShortName, PlatformLongName, PlatformType,
PlatformCharacteristic*, Instrument*)>
<!ELEMENT PlatformShortName (#PCDATA)>
<!ELEMENT PlatformLongName (#PCDATA)>
<!ELEMENT PlatformType (#PCDATA)>

<!ELEMENT PlatformCharacteristic (PlatformCharacteristicName,
PlatformCharacteristicDescription, PlatformCharacteristicDataType,
PlatformCharacteristicUnit?, PlatformCharacteristicValue)>
<!ELEMENT PlatformCharacteristicName (#PCDATA)>
<!ELEMENT PlatformCharacteristicDescription (#PCDATA)>
<!ELEMENT PlatformCharacteristicDataType (#PCDATA)>
<!ELEMENT PlatformCharacteristicUnit (#PCDATA)>
<!ELEMENT PlatformCharacteristicValue (#PCDATA)>

<!ELEMENT Instrument (InstrumentShortName, InstrumentLongName?,
InstrumentTechnique?, NumberOfSensors?, InstrumentCharacteristic*, Sensor*,
OperationMode*)>
<!ELEMENT InstrumentShortName (#PCDATA)>
<!ELEMENT InstrumentLongName (#PCDATA)>
<!ELEMENT InstrumentTechnique (#PCDATA)>
<!ELEMENT NumberOfSensors (#PCDATA)>

<!ELEMENT InstrumentCharacteristic (InstrumentCharacteristicName,
InstrumentCharacteristicDescription, InstrumentCharacteristicDataType,
InstrumentCharacteristicUnit?, InstrumentCharacteristicValue)>
<!ELEMENT InstrumentCharacteristicName (#PCDATA)>
<!ELEMENT InstrumentCharacteristicDescription (#PCDATA)>
<!ELEMENT InstrumentCharacteristicDataType (#PCDATA)>
<!ELEMENT InstrumentCharacteristicUnit (#PCDATA)>

```

```

<!ELEMENT InstrumentCharacteristicValue (#PCDATA)>

<!ELEMENT Sensor (SensorShortName, SensorLongName?, SensorTechnique?,
SensorCharacteristic*)>
<!ELEMENT SensorShortName (#PCDATA)>
<!ELEMENT SensorLongName (#PCDATA)>
<!ELEMENT SensorTechnique (#PCDATA)>

<!ELEMENT SensorCharacteristic (SensorCharacteristicName,
SensorCharacteristicDescription, SensorCharacteristicDataType,
SensorCharacteristicUnit?, SensorCharacteristicValue)>
<!ELEMENT SensorCharacteristicName (#PCDATA)>
<!ELEMENT SensorCharacteristicDescription (#PCDATA)>
<!ELEMENT SensorCharacteristicDataType (#PCDATA)>
<!ELEMENT SensorCharacteristicUnit (#PCDATA)>
<!ELEMENT SensorCharacteristicValue (#PCDATA)>
<!ELEMENT OperationMode (#PCDATA)>

<!ELEMENT AdditionalAttributes (AdditionalAttributeDataType,
AdditionalAttributeDescription, AdditionalAttributeName,
MeasurementResolution?, ParameterRangeBegin?, ParameterRangeEnd?,
ParameterUnitsOfMeasure?, ParameterValueAccuracy?, ValueAccuracyExplanation?,
ParameterValue*)>
<!ELEMENT AdditionalAttributeDataType (#PCDATA)>
<!ELEMENT AdditionalAttributeDescription (#PCDATA)>
<!ELEMENT AdditionalAttributeName (#PCDATA)>
<!ELEMENT MeasurementResolution (#PCDATA)>
<!ELEMENT ParameterRangeBegin (#PCDATA)>
<!ELEMENT ParameterRangeEnd (#PCDATA)>
<!ELEMENT ParameterUnitsOfMeasure (#PCDATA)>
<!ELEMENT ParameterValueAccuracy (#PCDATA)>
<!ELEMENT ValueAccuracyExplanation (#PCDATA)>
<!ELEMENT ParameterValue (#PCDATA)>

<!-- List of browse granules that are related to this collection -->
<!ELEMENT BrowseProduct (BrowseGranuleId*)>
<!ELEMENT BrowseGranuleId (#PCDATA)>

<!ELEMENT CSDDTDescription (PrimaryCSDDT, Implementation?, CSDDTComments?,
IndirectReference?)>
<!ELEMENT PrimaryCSDDT (#PCDATA)>
<!ELEMENT Implementation (#PCDATA)>
<!ELEMENT CSDDTComments (#PCDATA)>
<!ELEMENT IndirectReference (#PCDATA)>

<!ELEMENT Locality (LocalityType, LocalityDescription?)>
<!ELEMENT LocalityType (#PCDATA)>
<!ELEMENT LocalityDescription (#PCDATA)>

<!ELEMENT CollReview (ScienceReviewDate, ScienceReviewStatus,
FutureReviewDate?)>
<!ELEMENT ScienceReviewDate (#PCDATA)>
<!ELEMENT ScienceReviewStatus (#PCDATA)>
<!ELEMENT FutureReviewDate (#PCDATA)>

<!ELEMENT Documents (Document+)>
<!ELEMENT Document (DocumentType?, DocumentURL?, DocumentURLComment?)>
<!ELEMENT DocumentType (#PCDATA)>

```



```

<!ELEMENT DocumentURL (#PCDATA)>
<!ELEMENT DocumentURLComment (#PCDATA)>

<!ELEMENT CollectionAssociation (AssociatedShortName, AssociatedVersionId,
CollectionType, CollectionUse1?, CollectionUse2?)>
<!ELEMENT AssociatedShortName (#PCDATA)>
<!ELEMENT AssociatedVersionId (#PCDATA)>
<!ELEMENT CollectionType (#PCDATA)>
<!ELEMENT CollectionUse1 (#PCDATA)>
<!ELEMENT CollectionUse2 (#PCDATA)>

<!ELEMENT AnalysisSource (AnalysisType, AnalysisShortName, AnalysisLongName?,
AnalysisTechnique?)>
<!ELEMENT AnalysisType (#PCDATA)>
<!ELEMENT AnalysisShortName (#PCDATA)>
<!ELEMENT AnalysisLongName (#PCDATA)>
<!ELEMENT AnalysisTechnique (#PCDATA)>

<!ELEMENT Campaign (CampaignShortName, CampaignLongName?, CampaignStartDate?,
CampaignEndDate?)>
<!ELEMENT CampaignShortName (#PCDATA)>
<!ELEMENT CampaignLongName (#PCDATA)>
<!ELEMENT CampaignStartDate (#PCDATA)>
<!ELEMENT CampaignEndDate (#PCDATA)>

```

#### 4.15.2.13.2 BMGTGranuleMetadata.dtd

```

<!ELEMENT GranuleMetaDataFile (DTDVersion, DataCenterId, TemporalCoverage,
GranuleURMetaData*)>

<!-- Version identifier of the DTD used to generate the file -->
<!ELEMENT DTDVersion (#PCDATA)>

<!-- DataCenterId of the site that stores this metadata (e.g., LP DAAC) -->
<!ELEMENT DataCenterId (#PCDATA)>

<!-- the start and end dates of this MetaDataFile (YYYY-MM-DD) -->
<!ELEMENT TemporalCoverage (StartDate, EndDate)>
<!ELEMENT StartDate (#PCDATA)>
<!ELEMENT EndDate (#PCDATA)>

<!ELEMENT GranuleURMetaData
(GranuleUR, DbID?, InsertTime?, LastUpdate?, DeleteTime?, CollectionMetaData?,
ECSDDataGranule?, PGEVersionClass?, (RangeDateTime | SingleDateTime)?,
SpatialDomainContainer?, OrbitCalculatedSpatialDomain?, MeasuredParameter?,
ProcessingQA?, StorageMediumClass?, Review?, Platform*, AnalysisSource*,
Campaign*, PSAs?, InputGranule?, BrowseProduct?, PHProduct?, QAProduct?,
AlgorithmPackage*, AncillaryInputGranules?)>
<!ELEMENT GranuleUR (#PCDATA)>
<!ELEMENT DbID (#PCDATA)>
<!ELEMENT InsertTime (#PCDATA)>
<!ELEMENT LastUpdate (#PCDATA)>
<!ELEMENT DeleteTime (#PCDATA)>

<!ELEMENT CollectionMetaData (ShortName, VersionID)>
<!ELEMENT ShortName (#PCDATA)>
<!ELEMENT VersionID (#PCDATA)>

```

```

<!ELEMENT ECSDDataGranule
(SizeMBECSDDataGranule, ReprocessingPlanned?, ReprocessingActual?,
LocalGranuleID?, DayNightFlag?, ProductionDateTime, LocalVersionID?)>
<!ELEMENT SizeMBECSDDataGranule (#PCDATA)>
<!ELEMENT ReprocessingPlanned (#PCDATA)>
<!ELEMENT ReprocessingActual (#PCDATA)>
<!ELEMENT LocalGranuleID (#PCDATA)>
<!ELEMENT DayNightFlag (#PCDATA)>
<!ELEMENT ProductionDateTime (#PCDATA)>
<!ELEMENT LocalVersionID (#PCDATA)>

<!ELEMENT PGEVersionClass (PGEVersion)>
<!ELEMENT PGEVersion (#PCDATA)>

<!ELEMENT RangeDateTime
(RangeEndingTime, RangeEndingDate, RangeBeginningTime, RangeBeginningDate)>
<!ELEMENT RangeEndingTime (#PCDATA)>
<!ELEMENT RangeEndingDate (#PCDATA)>
<!ELEMENT RangeBeginningTime (#PCDATA)>
<!ELEMENT RangeBeginningDate (#PCDATA)>

<!ELEMENT SingleDateTime (TimeOfDay, CalendarDate)>
<!ELEMENT TimeOfDay (#PCDATA)>
<!ELEMENT CalendarDate (#PCDATA)>

<!ELEMENT SpatialDomainContainer
(GranuleLocality*, VerticalSpatialDomain*, HorizontalSpatialDomainContainer?)>

<!ELEMENT GranuleLocality (LocalityValue)>
<!ELEMENT LocalityValue (#PCDATA)>

<!ELEMENT VerticalSpatialDomain (VerticalSpatialDomainContainer)>

<!ELEMENT VerticalSpatialDomainContainer
(VerticalSpatialDomainType, VerticalSpatialDomainValue)>
<!ELEMENT VerticalSpatialDomainType (#PCDATA)>
<!ELEMENT VerticalSpatialDomainValue (#PCDATA)>

<!ELEMENT HorizontalSpatialDomainContainer
(ZoneIdentifierClass?, (Point | Circle | BoundingRectangle | GPolygon |
Global))>

<!ELEMENT ZoneIdentifierClass (ZoneIdentifier)>
<!ELEMENT ZoneIdentifier (#PCDATA)>

<!ELEMENT Point (PointLongitude, PointLatitude)>
<!ELEMENT PointLongitude (#PCDATA)>
<!ELEMENT PointLatitude (#PCDATA)>

<!ELEMENT Circle (CenterLatitude, CenterLongitude, Radius, RadiusUnits)>
<!ELEMENT CenterLatitude (#PCDATA)>
<!ELEMENT CenterLongitude (#PCDATA)>
<!ELEMENT Radius (#PCDATA)>
<!ELEMENT RadiusUnits (#PCDATA)>

<!ELEMENT BoundingRectangle
(WestBoundingCoordinate, NorthBoundingCoordinate, EastBoundingCoordinate,
SouthBoundingCoordinate)>

```

```

<!ELEMENT WestBoundingCoordinate (#PCDATA)>
<!ELEMENT NorthBoundingCoordinate (#PCDATA)>
<!ELEMENT EastBoundingCoordinate (#PCDATA)>
<!ELEMENT SouthBoundingCoordinate (#PCDATA)>

<!ELEMENT GPolygon (Boundary)+>
<!ELEMENT Boundary (Point, Point, Point, Point*)>

<!ELEMENT Global EMPTY>

<!ELEMENT OrbitCalculatedSpatialDomain
(OrbitCalculatedSpatialDomainContainer)+>

<!ELEMENT OrbitCalculatedSpatialDomainContainer
(OrbitalModelName?, OrbitNumber?, OrbitRange?, EquatorCrossingLongitude,
EquatorCrossingDate, EquatorCrossingTime)>

<!ELEMENT OrbitalModelName (#PCDATA)>
<!ELEMENT OrbitNumber (#PCDATA)>

<!ELEMENT OrbitRange (StartOrbitNumber, StopOrbitNumber)>
<!ELEMENT StartOrbitNumber (#PCDATA)>
<!ELEMENT StopOrbitNumber (#PCDATA)>

<!ELEMENT EquatorCrossingLongitude (#PCDATA)>
<!ELEMENT EquatorCrossingDate (#PCDATA)>
<!ELEMENT EquatorCrossingTime (#PCDATA)>

<!ELEMENT MeasuredParameter (MeasuredParameterContainer)+>
<!ELEMENT MeasuredParameterContainer (ParameterName, QAStats?, QAFlags?)>
<!ELEMENT ParameterName (#PCDATA)>

<!ELEMENT QAStats
(QAPercentMissingData, QAPercentOutOfBoundsData?, QAPercentInterpolatedData?,
QAPercentCloudCover?)>
<!ELEMENT QAPercentMissingData (#PCDATA)>
<!ELEMENT QAPercentOutOfBoundsData (#PCDATA)>
<!ELEMENT QAPercentInterpolatedData (#PCDATA)>
<!ELEMENT QAPercentCloudCover (#PCDATA)>

<!ELEMENT QAFlags
(AutomaticQualityFlag?, AutomaticQualityFlagExplanation?,
OperationalQualityFlag?, OperationalQualityFlagExplanation?,
ScienceQualityFlag?, ScienceQualityFlagExplanation?)>
<!ELEMENT AutomaticQualityFlag (#PCDATA)>
<!ELEMENT AutomaticQualityFlagExplanation (#PCDATA)>
<!ELEMENT OperationalQualityFlag (#PCDATA)>
<!ELEMENT OperationalQualityFlagExplanation (#PCDATA)>
<!ELEMENT ScienceQualityFlag (#PCDATA)>
<!ELEMENT ScienceQualityFlagExplanation (#PCDATA)>

<!ELEMENT ProcessingQA (ProcessingQAContainer)+>

<!ELEMENT ProcessingQAContainer (ProcessingQADescription,
ProcessingQAAttribute)>
<!ELEMENT ProcessingQADescription (#PCDATA)>
<!ELEMENT ProcessingQAAttribute (#PCDATA)>

```

```

<!ELEMENT StorageMediumClass (StorageMedium)+>
<!ELEMENT StorageMedium (#PCDATA)>

<!ELEMENT Review (ReviewContainer)+>
<!ELEMENT ReviewContainer (ScienceReviewStatus, ScienceReviewDate,
FutureReviewDate?)>
<!ELEMENT ScienceReviewStatus (#PCDATA)>
<!ELEMENT ScienceReviewDate (#PCDATA)>
<!ELEMENT FutureReviewDate (#PCDATA)>

<!ELEMENT Platform (PlatformShortName,Instrument*)>
<!ELEMENT PlatformShortName (#PCDATA)>

<!ELEMENT Instrument (InstrumentShortName, Sensor*, OperationMode*)>
<!ELEMENT InstrumentShortName (#PCDATA)>
<!ELEMENT OperationMode (#PCDATA)>

<!ELEMENT Sensor (SensorShortName, SensorCharacteristic*)>
<!ELEMENT SensorShortName (#PCDATA)>

<!ELEMENT SensorCharacteristic (SensorCharacteristicName,
SensorCharacteristicValue)>
<!ELEMENT SensorCharacteristicName (#PCDATA)>
<!ELEMENT SensorCharacteristicValue (#PCDATA)>

<!ELEMENT AnalysisSource (AnalysisShortName)>
<!ELEMENT AnalysisShortName (#PCDATA)>

<!ELEMENT Campaign (CampaignShortName)>
<!ELEMENT CampaignShortName (#PCDATA)>

<!ELEMENT PSAs (PSA+)>
<!ELEMENT PSA (PSAName, PSAValue+)>
<!ELEMENT PSAName (#PCDATA)>
<!ELEMENT PSAValue (#PCDATA)>

<!ELEMENT InputGranule (InputPointer+)>
<!ELEMENT InputPointer (#PCDATA)>

<!-- List of browse granules that are related to this granule -->
<!ELEMENT BrowseProduct (BrowseGranuleId+)>
<!ELEMENT BrowseGranuleId (#PCDATA)>

<!-- List of production history granules that are related to this granule -->
<!ELEMENT PHProduct (PHGranuleId+)>
<!ELEMENT PHGranuleId (#PCDATA)>

<!-- List of QA granules that are related to this granule -->
<!ELEMENT QAProduct (QAGranuleId+)>
<!ELEMENT QAGranuleId (#PCDATA)>

<!ELEMENT AlgorithmPackage (AlgorithmPackageName, AlgorithmPackageVersion,
AlgorithmPackageMaturityCode, AlgorithmPackageAcceptDate, DeliveryPurpose,
PGENAME, PGEVersion, PGEIdentifier, PGEFunction, PGEDateLastModified,
SWVersion, SWDateLastModified, SSAPComponent*)>
<!ELEMENT AlgorithmPackageName (#PCDATA)>
<!ELEMENT AlgorithmPackageVersion (#PCDATA)>

```

```

<!ELEMENT AlgorithmPackageMaturityCode (#PCDATA)>
<!ELEMENT AlgorithmPackageAcceptDate (#PCDATA)>
<!ELEMENT DeliveryPurpose (#PCDATA)>
<!ELEMENT PGName (#PCDATA)>
<!ELEMENT PGEIdentifier (#PCDATA)>
<!ELEMENT PGEFunction (#PCDATA)>
<!ELEMENT PGDateLastModified (#PCDATA)>
<!ELEMENT SWVersion (#PCDATA)>
<!ELEMENT SWDateLastModified (#PCDATA)>

<!ELEMENT SSAPComponent (ComponentType, ComponentName,
SSAPAlgorithmPackageName, SSAPInsertDate)>
<!ELEMENT ComponentType (#PCDATA)>
<!ELEMENT ComponentName (#PCDATA)>
<!ELEMENT SSAPAlgorithmPackageName (#PCDATA)>
<!ELEMENT SSAPInsertDate (#PCDATA)>

<!ELEMENT AncillaryInputGranules (AncillaryInputGranule+)>
<!ELEMENT AncillaryInputGranule (AncillaryInputType, AncillaryInputPointer)>
<!ELEMENT AncillaryInputType (#PCDATA)>
<!ELEMENT AncillaryInputPointer (#PCDATA)>

```

#### 4.15.2.13.3 BMGTBrowseMetadata.dtd

```

<!ELEMENT BrowseReferenceFile (DTDVersion, DataCenterId, TemporalCoverage,
BrowseCrossReference*)>

<!-- Version identifier of the DTD used to generate the file -->
<!ELEMENT DTDVersion (#PCDATA)>

<!-- DataCenterId of the site that stores this metadata (e.g., LP DAAC-EMD) -->
>
<!ELEMENT DataCenterId (#PCDATA)>

<!-- the start and end dates of this MetaDataFile (YYYYDDD) -->
<!ELEMENT TemporalCoverage (StartDate, EndDate)>
<!ELEMENT StartDate (#PCDATA)>
<!ELEMENT EndDate (#PCDATA)>

<!ELEMENT BrowseCrossReference (GranuleUR, BrowseGranuleId?, InsertTime?,
LastUpdate?, DeleteTime?, InternalFileName, BrowseDescription?, BrowseSize?)>
<!ELEMENT GranuleUR (#PCDATA)>
<!ELEMENT BrowseGranuleId (#PCDATA)>
<!ELEMENT InsertTime (#PCDATA)>
<!ELEMENT LastUpdate (#PCDATA)>
<!ELEMENT DeleteTime (#PCDATA)>
<!ELEMENT InternalFileName (#PCDATA)>
<!ELEMENT BrowseDescription (#PCDATA)>
<!ELEMENT BrowseSize (#PCDATA)>

```

#### 4.15.2.13.4 BMGTValidMetadata.dtd

```

<!ELEMENT ValidFile (DTDVersion, DataCenterId, TemporalCoverage,
DictionaryAttribute+, KeywordValid+)>

<!-- Version identifier of the DTD used to generate the file -->
<!ELEMENT DTDVersion (#PCDATA)>

```

```

<!-- DataCenterId of the site that stores the metadata (e.g. LP DAAC-EMD) -->
<!ELEMENT DataCenterId (#PCDATA)>

<!-- The start and end dates of this MetaDataFile (YYYYDDD) -->
<!ELEMENT TemporalCoverage (StartDate, EndDate)>
<!ELEMENT StartDate (#PCDATA)>
<!ELEMENT EndDate (#PCDATA)>

<!-- Attributes and their Data Types -->
<!ELEMENT DictionaryAttribute (QualifiedAttrName, Type, Length, RuleText*)>
<!ELEMENT QualifiedAttrName (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Length (#PCDATA)>
<!ELEMENT RuleText (#PCDATA)>

<!-- Keyword Attributes and their Domain Values -->
<!ELEMENT KeywordValid (DisciplineKeyword, TopicKeyword, TermKeyword,
VariableKeyword?, ParameterKeyword?)>
<!ELEMENT DisciplineKeyword (#PCDATA)>
<!ELEMENT TopicKeyword (#PCDATA)>
<!ELEMENT TermKeyword (#PCDATA)>
<!ELEMENT VariableKeyword (#PCDATA)>
<!ELEMENT ParameterKeyword (#PCDATA)>

```

#### 4.15.2.13.5 BMGTUpdateMetadata.dtd

```

<!ELEMENT ProviderAccountService (UpdateMetadata)>
<!--UpdateMetadata can update a single collection, multiple collections, a
single granule, or multiple granules in one transaction. Each update allows
the addition of new metadata-->
<!ELEMENT UpdateMetadata (Collection*, Granule*)>
<!ELEMENT Collection (Target+, (Add | Update | Delete)+)>
<!ELEMENT Granule (Target+, (Add | Update | Delete)+)>
<!-- Target+ allows the same change to be made to several different granules
or collections simultaneously. This is especially useful for bulk deletions
of OnlineURLs. -->
<!ELEMENT Target (ID, ProviderLastUpdateDateTime, SaveDateTimeFlag?)>
<!-- SaveDateTimeFlag is the flag that allows echo to update the last update
date time for Target. The default is SAVE -->
<!ELEMENT Add (QualifiedTag, MetadataValue)>
<!ELEMENT Update (QualifiedTag, MetadataValue)>
<!ELEMENT Delete (QualifiedTag+)>
<!ELEMENT QualifiedTag (#PCDATA)>
<!ELEMENT MetadataValue (#PCDATA)>
<!ELEMENT ProviderLastUpdateDateTime (#PCDATA)>
<!ELEMENT SaveDateTimeFlag (SAVE | DONTSAVE)>
<!ELEMENT SAVE EMPTY>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT DONTSAVE EMPTY>

```

## 4.16 OGC-ECHO Adaptor (OEA) Subsystem Overview

The OGC-ECHO Adaptor (OEA) provides a mechanism to allow the Earth Science Gateway Portal and other OGC clients to perform OGC catalogue searches against ECHO holdings. This is presented in an OGC-compliant fashion, which is accomplished via servicing Z39.50 GEO profile catalog requests from the Earth Science Gateway Portal.

The OEA complies with version 2 of the OGC Catalog Services Specification ([http://portal.opengis.org/files/?artifact\\_id=5929&version=1](http://portal.opengis.org/files/?artifact_id=5929&version=1)) using the Z39.50 binding. This specification defines the following mandatory requests.

Housekeeping Requests:

- Initiate Session
- Terminate Session
- Status
- Cancel

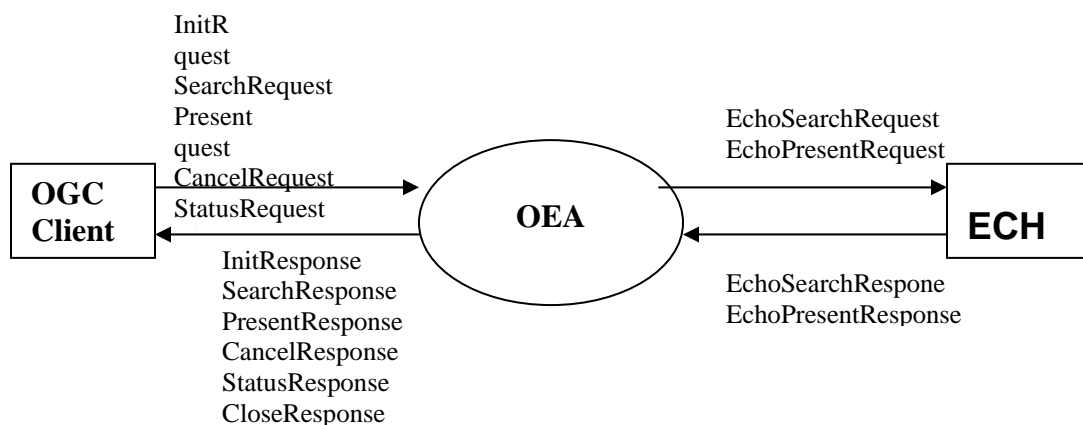
Catalog requests will require interaction with the ECHO Application Programming Interface. These will be rendered into ECHO client API requests by the OEA. The results of those ECHO API requests will be rendered into the counterpart ESG Portal catalog results. Those requests being,

- Search
- Present

Consequently, the OEA will comply with version 6.0 of the ECHO API (<http://api.echo.eos.nasa.gov/echo/About.jsp>)

### OGC-ECHO Adaptor (OEA) Context

Figure 4.16-1 describes the interaction between the ESG Portal, OEA and ECHO API.



**Figure 4.16-1. OEA Context Diagram**

Note that the OEA will be subject to interrogation of two distinct types,

- On demand searches from individual users. Individual search and present operations for distinct inventory.
- Harvesting operations from NSDI (National Spatial Data Infrastructure) clients. Broad search and present operations with the intent of replicating all or part of the NDSI node's data holdings. Harvesting operations are subject to repetition in order to keep the client's inventory copy up to date.

The ESG, for example, can operate in both modes.

Table 4.16-1 provides descriptions of the interface events shown in the OEA context diagram.

**Table 4.16-1. OEA Interface Events (1 of 2)**

Event	Interface Event Description
InitRequest	OEA receives Init Request from Client to create a new session.
InitResponse	OEA creates a new session and sends back a valid init response to notify client that a session has been generated.
SearchRequest	OEA receives a Search Request from Client; it could involve granule, collection, granule and collection, temporal and so on.
EchosearchRequest	OEA translates the Search to XML format and submits it to ECHO on behave of a user.
CloseRequest	OEA receives Close Request from Client to terminate a connection.
CloseResponse	OEA cleans up a session, send back a Close Response and terminates the connection.
PresentRequest	OEA receives a Present Request from Client with some resultSetName.



**Table 4.16-1. OEA Interface Events (2 of 2)**

Event	Interface Event Description
EchoPresentRequest	OEA translates the Present to XML format and submits it to ECHO on behalf of a user.
StatusRequest	OEA receives a Status Request during a Search/Present.
StatusResponse	OEA sends back a Status Response only if the Search/Present request is not in a terminal state (Canceled, Failed, Done).
CancelRequest	OEA receives Cancel Request during a Search/Present to terminate the request.
CancelResponse	OEA stops the Search/Present and sends back a Cancel response only if the Search/Present is not in a terminal state (Canceled, Failed, Done).

#### **4.16.1 OGC-ECHO Adaptor Architecture**

The OEA is functionally divided into three packages -- *z3950serverfacade package*, *translation package*, and *Echoclientfacade package*. The reason for this decoupling of functionality allows reuse of each element in possible future adaptor development. Certainly, Z39.50 GEO is not the only protocol compliant with the OGC Catalogue Service Specification (http and xml may become prevalent in the future).

- **z3950serverfacade package**

The z3950serverfacade is responsible for session management between OEA and OGC clients. It also validates and parses all incoming z3950 requests (Init, Search, Present, Status, Cancel and Close), calls the translation package to translate all the Search/Present requests into XML formats, and sends them to the echoclientfacade package for processing. After receiving responses from ECHO, it calls the translation package again to translate the responses into z3950 Search/Present responses and eventually sends them back to the client. For any other type of requests (Init, Status, Cancel, Close), the z3950serverfacade also generates responses and sends them back to the client.

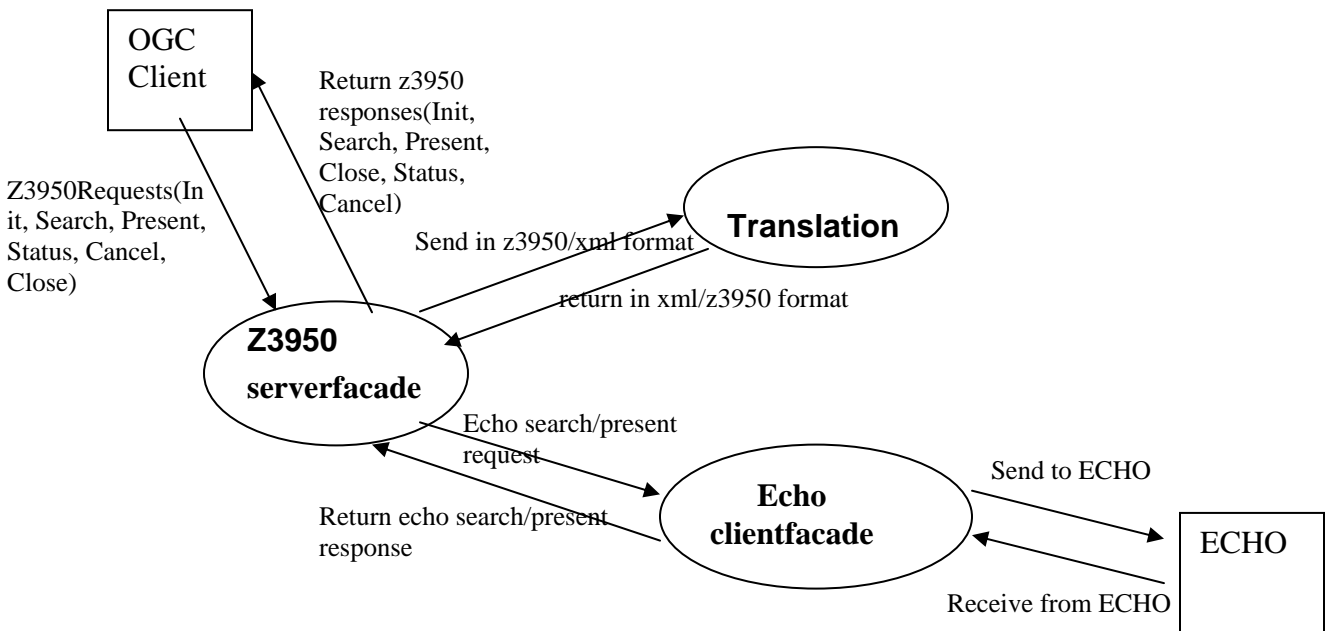
- **translation package**

All requests that are submitted to OEA from ESG are in z3950 format, however, ECHO handles all its requests and responses in XML format. This is when the translation package comes in. Equipped with style sheets for both Search and Present, the translation package handles the crucial translation of requests and responses between z3950 and XML formats. It ensures no loss of information between the requests that come in from ESG and the requests that are sent to ECHO and vice versa.

- **echoclientfacade package**

The Echoclientfacade handles session management between OEA and ECHO by creating a session with a valid username and password per ECHO request. It also manages all Echo request and response objects, sends and receives requests to and from ECHO.

Figure 4.16-2 describes the OEA processes to handle a request and Table 4.16-2 provides descriptions of the events in Figure 4.16-2.



**Figure 4.16-2. OEA Architecture**

#### 4.16.1.1 OEA Process Interface Descriptions

Table 4.16-2 provides descriptions of the interface processes shown in the OEA architecture.

**Table 4.16-2. OEA Processes Interface Events (1 of 3)**

Event	Event Frequency	Interface	Initiated By	Event Description
z3950Requests (Init, Search, Present, Status, Cancel, Close)	One per request	Process: EcOwOea Library: Z3950serverfacade Class: Server Session InitService CloseService TRCService SearchService PresentService	Process: OGCCClient	The OEA receives requests from Clients to do one of the following: Initiate a Session Search Echo Inventory Present a resultSet Request status of a particular Search/Present Cancel a Search/Present Close a Session OEA then validates and parses the request

**Table 4.16-2. OEA Processes Interface Events (2 of 3)**

Event	Event Frequency	Interface	Initiated By	Event Description
Send in z3950/xml format	One call to translator per echo request	<i>Process:</i> EcOwOea <i>Library:</i> translation <i>Class:</i> ServiceTranslator SearchTranslator PresentTranslator XMLGenerator	<i>Process:</i> EcOwOea <i>Library:</i> Z3950serverfacade <i>Class:</i> SearchService PresentService	For any Search/Present request, OEA calls the translator to translate z3950 requests into XML format, so they later can be sent to ECHO. When OEA receives request from ECHO, it will also call the translator to translate the responses back to z3950 responses.
Return in xml/z3950format	One z3950response per request	<i>Process:</i> EcOwOea <i>Library:</i> Z3950Serverfacade <i>Class:</i> SearchService PresentService	<i>Process:</i> EcOwOea <i>Library:</i> translation <i>Class:</i> SearchTranslator PresentTranslator	After applying style sheets, OEA passes the newly formatted request/response back to z3950serverfacade package to get ready for ECHO
Echo Search/present Request	One per z3950 request	<i>Process:</i> EcOwOea <i>Library:</i> echoclientfacade <i>Class:</i> ECHOSearchRequest ECHOPresentRequest Session SearchService PresentService	<i>Process:</i> EcOwOea <i>Library:</i> Z3950serverfacade <i>Class:</i> SearchService PresentService	After translation, OEA creates the ECHO Search/Present object, and establishes a session to ECHO per request
Send to ECHO	One connection per request	<i>Process:</i> ECHO	<i>Process:</i> EcOwOea <i>Library:</i> echoclientfacade <i>Class:</i> Session SessionPool	OEA sends the ECHO Search/Present request to ECHO and awaits responses.

**Table 4.16-2. OEA Processes Interface Events (3 of 3)**

Event	Event Frequency	Interface	Initiated By	Event Description
Receive from ECHO	One ECHO response per request	<i>Process:</i> EcOwOea <i>Library:</i> echoclientfacade <i>Class:</i> Session SessionPool	<i>Process:</i> ECHO	ECHO returns the Search/Present result
Return Echo Search/Present Response	One per Echo response	<i>Process:</i> EcOwOea <i>Library:</i> Z3950serverfacade <i>Class:</i> SearchService PresentService	<i>Process:</i> EcOwOea <i>Library:</i> echoclientfacade <i>Class:</i> ECHOSearchRequest ECHOPresentRequest	Echoclientfacade returns the ECHO Search/Present responses back to z3950serverfacade .Z3950serverfacade then calls translation to translate the responses back into z3950 responses.
Z3950Response (Init, Search, Present, Status, Cancel, Close)	One per Z3950 request	<i>Process:</i> OGC Clients	<i>Process:</i> EcOwOea <i>Library:</i> Z3950serverfacade <i>Class:</i> Server Session InitService CloseService TRCService SearchService PresentService	The OEA finally send z3950 responses back to the Client (Note for the requests not going thru ECHO: Init, Close, Status and Cancel, z3950serverfacade will construct the response accordingly right after validation and parsing of the incoming request)

#### 4.16.1.2 Use of COTS in the OEA

- **Java 1.4.2**

The Java1.4.2 class libraries are used by the OEA to provide basic functions and objects such as strings and collections. The Java libraries must be installed with the OEA software for any of the OEA processes to run.

- **Jzkit**

The Jzkit class libraries are a pure java toolset for building advanced search and retrieve applications. It assists in implementing Z3950 standard in pure java environment. The Jzkit libraries must be installed with OEA software for any interactions between OGC and OEA.

- **Jdom**

The Jdom class libraries enables the use of XSL style sheets to translate any Z3950 requests to XML formatted ECHO requests and vice versa. It has to be installed to allow proper translation between OEA and ECHO.

- **Echo Client Toolkit**

Allows OEA to query ECHO's metadata clearinghouse for data sets and granules by using a set of Java classes that interact with the ECHO system. It has to be installed to allow proper communication between OEA and ECHO.

- **Apache Soap**

Apache SOAP ("Simple Object Access Protocol") is an implementation of the SOAP submission to W3C. It is based on, and supersedes, the IBM SOAP4J implementation. All requests are submitted to ECHO through a SOAP mechanism. It has to be installed to allow proper communication between OEA and ECHO.

The current plan is to deploy it on ECHO hardware but it can be deployed on EMD hardware. It is a single component, EcOwOgcEchoAdaptor. It can be deployed to e0ogp01 only according to the sitemap, the same as the other DOWS software components.

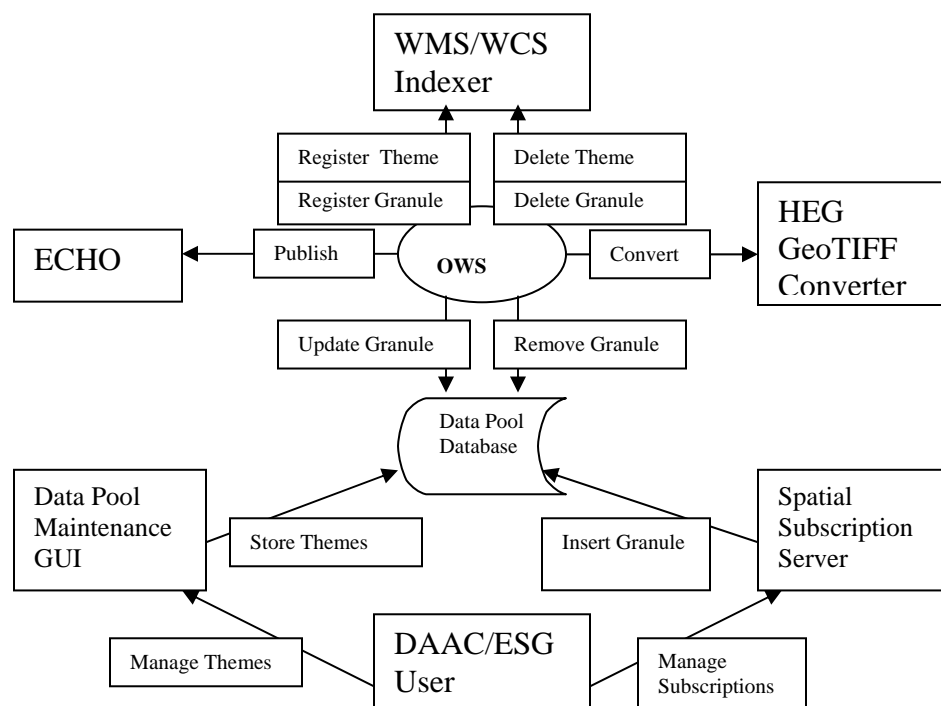
## 4.17 Open Geospatial Consortium (OGC) Web Services (OWS) Functional Overview

The Open Geospatial Consortium (OGC) Web Services (OWS) subsystem is the means by which a subset of Data Pool granules is made visible to the Web Coverage Service (WCS) and/or Web Mapping Service (WMS).

The OWS subsystem is comprised of three logical components: data enabling, data conversion, and data synchronization. Data enabling allows for the creation of OWS granules by allowing the user to create OWS themes, which will be associated with Data Pool insert actions for subscriptions. Data conversion allows for the preconversion of OWS granules to GeoTIFF format. Data synchronization is concerned with keeping the list of OWS granules in the Data Pool database synchronized with the data objects registered with the WCS/WMS indexer.

### Open Geospatial Consortium (OGC) Web Services (OWS) Context

Figure 4.17-1 is the OWS context diagram, showing how OWS interfaces with external objects. Table 4.17-1 provides descriptions of the interface events in the OWS context diagram.



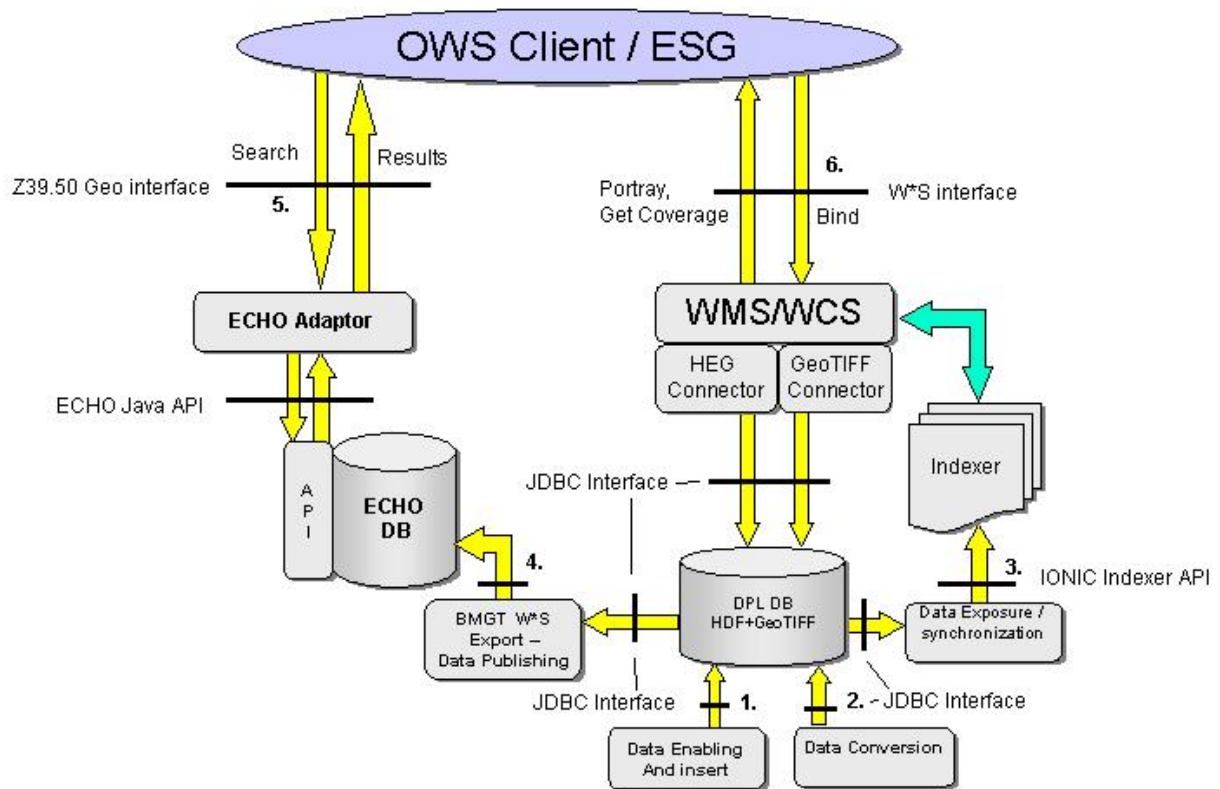
**Figure 4.17-1. OWS Context Diagram**

**Table 4.17-1. OWS Interface Events**

<b>Event</b>	<b>Interface Event Description</b>
Manage Themes	A user can create a theme using the Data Pool Maintenance GUI. The theme can be designated an OWS theme by specifying a service (WCS, WMS, or both) and, optionally, by setting a preconvert flag, indicating that granules for this theme should be preconverted to GeoTIFF format prior to being registered with the WMS or WCS. The user can also update or delete themes.
Store Themes	The Data Pool Maintenance GUI will record information about themes that have been created, updated, or deleted in the Data Pool database.
Manage Subscriptions	A user can create a subscription, using the Spatial Subscription Server, and associate a Data Pool insert action with the subscription. This action can have an associated theme, which could be an OWS theme. The user can also update or delete subscriptions.
Insert Granule	When a new granule, matching a subscription with a Data Pool action, arrives in the system, the Spatial Subscription Server will add it to the Data Pool insert action queue in the Data Pool database.
Register Granule	OWS will register an OWS granule with the WCS/WMS indexer.
Register Theme	OWS will register an OWS theme with the WCS/WMS indexer.
Update Granule	OWS will update the processing state of an OWS granule to “converted” after it has been preconverted to GeoTIFF format and to “indexed” after it has been registered with the WCS/WMS indexer.
Delete Granule	OWS will notify the WCS/WMS indexer when an OWS granule has been deleted from the Data Pool database.
Delete Theme	OWS will notify the WCS/WMS indexer when an OWS theme has been deleted from the Data Pool database.
Remove Granule	OWS will remove the granule from the list of deleted OWS granules after it has been deleted by the WCS/WMS indexer.
Convert	OWS will request that granules needing preconversion be converted to GeoTIFF format and will store information about the file location of the converted data in the Data Pool database.
Publish	Information about OWS granules is exported to the ECHO database by the OWS Binding Utility (OBU)

#### **4.17.1 OWS Architecture**

Figure 4.17-2 is the OWS architecture diagram, showing the principal OWS processes.



**Figure 4.17-2. OWS Architecture Diagram**

Table 4.17-2 provides descriptions of the processes shown in the OWS architecture diagram.



**Table 4.17-2. OWS Processes**

Process	Type	Hardware CI	COTS/ Developed	Functionality
Data Enabling	Perl Script	DPSHW	Developed	This is not an original process but rather was implemented as modifications to existing code. Data is enabled by creating an OWS theme, using the Data Pool Maintenance GUI, and by creating a subscription associated with that theme, using the Spatial Subscription Server GUI. When a new granule matches the subscription, the SSS runtime server will cause it to be inserted into the Data Pool.
Data Conversion	Java Executable	DPSHW	Developed	Data Conversion monitors the Data Pool database for new inventory, i.e. OWS granules, whose science data require preconversion to GeoTIFF format. It coordinates this conversion with the HEG server, advancing the processing state of converted OWS granules to 'converted' in the Data Pool database. It will also record in the database the location of the new 'GEO' files that are generated.
Data Synchronization	Java Executable	DPSHW	Developed	Data Synchronization monitors the Data Pool database for new OWS inventory. It then registers these granules with WMS/WCS via the Indexer API, advancing the processing state in the database to 'indexed'. OWS themes are also registered with the Indexer as aggregates. Synchronization also looks for OWS granules that have been deleted by Data Pool cleanup and notifies the Indexer of their deletion, after which it removes them from the list of deleted OWS granules.
Data Publishing	Java Executable	DPSHW	Developed	Data Publishing is represented by the OWS Binding Utility (OBU). OBU exports metadata for indexed OWS granules to the ECHO database, advancing their processing state in the Data Pool database to 'published'.
Data Pool Cleanup	Perl Script	DPSHW	Developed	Data Pool cleanup removes all files for expired granules from the Data Pool and all metadata for such granules from the Data Pool database. When an OWS granule is deleted, cleanup will add this information to the list of deleted OWS granules so that Data Synchronization can convey this information to the Indexer.
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE is where the Data Pool database resides.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

#### 4.17.1.1 OWS Process Interface Descriptions

Table 4.17-3 provides descriptions of the interface events shown in the OWS architecture diagram.

**Table 4.17-3. OWS Server Process Interface Events (1 of 2)**

Event	Event Frequency	Interface	Initiated by	Event Description
Get New Inventory	Once per execution	Stored procedure call	Data Conversion, Data Synchronization	The Data Pool database is searched for new OWS granules, i.e. new granules associated with an OWS theme. If the theme has preConvertEnabledFlag set, its granules need to be preconverted to GeoTIFF format. OWS granules are inserted into DGranuleOWS
GeoTIFF Conversion	Once per granule	HEG API	Data Conversion	A request for GeoTIFF conversion is submitted to the GeoTIFF converter (HEG server).
Reproject	Once per granule	WMS/WCS	Data Conversion	A converted granule is re-projected, if necessary, to the geographic projection.
Register Files	Once per granule	Database insert	Data Conversion	The location of the GeoTIFF file is inserted into DIFile.
Add theme	Zero or more times per GUI session	Stored procedure call	DPM GUI	A new theme can be added via the DPM GUI. It is designated as an OWS theme by specifying the service requested.
Get themes	Zero or more times per GUI session	Stored procedure call	DPM GUI	View the list of all themes or view the list of OWS themes.
Update theme	Zero or more times per GUI session	Stored procedure call	DPM GUI	Update the details of an existing theme.
Delete theme	Zero or more times per GUI session	Stored procedure call	DPM GUI	Delete a theme from the database.

**Table 4.17-3. OWS Server Process Interface Events (2 of 2)**

Event	Event Frequency	Interface	Initiated by	Event Description
Synchronize New Inventory	Once per execution	Stored procedure call + Java function call	Data Synchronization	Register themes, granules, and GeoTIFF files in the Indexer. Mark granules as 'indexed' in DIGranuleOWS and record GEO file locations in DIFile.
Synchronize Deleted Inventory	Once per execution	Stored procedure call + Java function call	Data Synchronization	Get deleted inventory from DPL database. Notify Indexer of deletions and await confirmation. Remove granules from deleted inventory list.
Synchronize Deleted Themes	Once per execution	Java function call	Data Synchronization	Notify indexer of deleted themes.
Get Indexed	Once per execution	Stored procedure call	Data Publishing (OBU)	Get list of OWS granules that have been successfully indexed and are ready for export to ECHO.
Get Deleted	Once per execution	Stored procedure call	Data Publishing (OBU)	Get list of OWS granules that are designated for cleanup.
Mark Published	Once per execution	Stored procedure call	Data Publishing (OBU)	Mark granules as 'published' in DIGranuleOWS if successfully exported to ECHO.

#### **4.17.1.2 OWS Data Stores**

Table 4.17-4 provides descriptions of the data found in the principal Sybase ASE data stores used by OWS. The three main types of data used by OWS processes are (1) granules, (2) files, and (3) themes. More detail on these and other data stores can be found in the Data Pool Database Design and Schema Specifications for the EMD Project.

**Table 4.17-4. OWS Data Stores**

<b>Data Store</b>	<b>Type</b>	<b>Functionality</b>
DIGranules	Granules	This is the principal table identifying granules which reside in the Data Pool.
DIThemes	Themes	This is the table identifying Data Pool themes. OWS themes are characterized by a non-null value in column serviceRequested. Also, the column preConvertEnabledFlag indicates whether granules associated with an OWS theme are to be preconverted prior to indexing.
DIGranuleThemeXref	Granules, Themes	A cross-reference table between themes and granules.
DIGranuleOWS	Granules	This table contains a list of active OWS granules in the Data Pool. The processingState column indicates the state of maturity of the granule: converted, indexed, or published. Inserts and updates to this table are performed by both Data Conversion and Data Synchronization. Data Publishing performs only updates. Data Pool cleanup performs deletes.
DIOWSDeletedGranules	Granules	This table contains a list of OWS granules that have expired. Inserts to this table are performed by Data Pool cleanup. Data Synchronization performs deletes.
DIFile	Files	This table contains information about the location of files in the Data Pool. A file in GeoTIFF format will have the value 'GEO' in the fileType column. Also, the concatenation of directoryPath and filename provides a complete description of the location of a GEO file.
DIFileSystems	Files	This table contains information about the location of file systems. It also contains flags indicating whether a file system is available for use and has available storage.
DICompressionAlgorithms	Files	This table contains information on the compression and decompression algorithms applied to a file.
DIFilesToBeDeleted	Files	This table contains a list of expired granules that Data Pool cleanup is about to delete. It is checked by Data Publishing, so that expired granules are not exported to the ECHO database.

This page intentionally left blank.

## 5. Limitations of Current Implementation

---

### 5.1 Data Server Subsystem

#### Science Data Server (SDSRV) CSCI

- **Operator GUI:** There is no support for re-installation of ESDTs from the GUI. Reinstallation of ESDTs is supported through a command line Unix Shell script interface. Corresponding data must be manually deleted from the Subscription Server and Data Dictionary.
- Metadata update services only support QA metadata.
- There is no persistence of Client requests, except for asynchronous acquire requests. Requests must be re-submitted.
- There is no support for Access Control List checking.

### 5.2 Planning Subsystem

#### Production Planning CSCI

- The Production Request Editor has not been optimized when creating Production Requests. The creation of large production requests with many input and output granules can take some time. This is mostly due to multiple database accesses; some database accesses may need to be replaced in the future with stored procedures to improve performance.
- There is no inter-DAAC planning at this time.

### 5.3 Infrastructure Subsystem

#### ASTER DAR Gateway CSCI

- The gateway itself does not extend the DAR functionality; but is limited by the functionality provided by the ASTER GDS through the API set.

#### E-mail Parser Gateway CSC

- E-mail Parser Gateway only handles Expedited Data Request.
- E-mail Parser Gateway only handles FTTPUSH for media type.

#### Subscription Server CSC/Subscription GUI

- The user can only subscribe to future granules.
- The Subscription Server validates String, long, double, date and time type qualifiers.

- Support for trigger persistence was merged in Release 5B.
- Subscription Server does not check security issues for updating its database tables. It assumes the client application does this.
- Under extreme rare situations, the Subscription Server might send two e-mail notifications for one subscription trigger.

#### **Machine-to-Machine Gateway CSC**

- The machine-to-machine gateway relies on ssh for the authentication of a connection.
- Each request must be associated with a MSS User ID. The operator has the choice of configuring a default MSS User ID for each MTMGW server. If there is no default for a server and the user does not provide a User ID in the request, the server rejects requests that do not specify a User ID.

## **5.4 MSS Subsystem**

#### **MCI CSCI – Security CSC**

- If an unauthorized user gains access to a host despite security measures, it is not detected until the next detection interval expires, since this is only checked periodically. (If the detection intervals were decreased, the system uses too much of its processing power for monitoring itself.)
- The security implementation requires the operator to perform security tasks such as running Crack manually.
- The configuration files for TCP Wrappers can become difficult to manage if multiple versions exist. If administrators setup a “back door,” the system security can be more easily compromised.

#### **MCI CSCI – Accountability CSC**

- There is no retry in place for Database updates or inserts and errors are logged as low priority.

#### **MCI CSCI – Trouble Ticketing CSC**

- Trouble Tickets forwarded from a DAAC to the SMC are set to a forwarded state in the DAAC. A manual process is necessary to receive notification that the Trouble Ticket has been closed at the SMC and is now closed at the DAAC.
- For an overall status of Trouble Tickets in the EMD system, reports must be run at each DAAC and forwarded via e-mail where they can be consolidated.

## **5.5 Product Distribution System (PDS) Subsystem (REMOVED)**

## **5.6 Spatial Subscription Server (SSS) Subsystem**

- Subscriptions can be entered only for future granules. It is not possible to order existing granules via subscription.
- Non-spatial qualification of a subscription is limited to five numeric/date qualifiers and fifty string qualifiers.
- Spatial qualification is limited to the specification of the coordinates of an overlapping latitude-longitude box.
- It is not possible to request Data Pool insertion for a granule when entering a subscription via the command line interface. The subscription must be entered via the GUI if Data Pool insertion is desired.

## **5.7 Open Geospatial Consortium (OGC) Web Services (OWS) Subsystem**

- For the initial release, only ten data types (ESDTs) are supported.
- The initial release will be to a single DAAC. (LP DAAC)
- Web Mapping and Web Coverage Service are limited to stitching up to 5 granules together.

## **5.8 Order Manager Server (OMS) Subsystem**

- Order Manager Server does not directly handle Secure Copy requests.
- All Secure Copy requests are automatically forwarded to Science Data Server.

## **5.9 Data Management (DMS) Subsystem**

No limitations.



This page intentionally left blank.

# Abbreviations and Acronyms

---

## A

ABC++	Document Generator used to provide class level detail
ACMHW	Access and Control Management Hardware (Configuration Item)
AD	Advertisement
ADC	Affiliated Data Center (National Oceanic and Atmospheric Administration only)
AGS	ASTER Ground System
AIT	Algorithm Integration and Test
AITHW	Algorithm Integration and Test Hardware (Configuration Item)
AI&T	Algorithm Integration and Test
AITTL	Algorithm Integration and Test Tools (Computer Software Configuration Item)
AM-1	See TERRA (spacecraft)
AMASS	Archival Management and Storage System  <i>Storage Management Software created by EMASS, which is the Raytheon Company, which in turn owns a part of Hughes Aircraft company, owns a wholly owned subsidiary of E-Systems Inc. E-Systems.</i>  <i>For more info you can visit: <a href="http://www.emass.com">http://www.emass.com</a></i>
AOI	Area of Interest
AOS	ASTER Operations Segment
AP	Algorithm Package
APC	Access/Process Coordinators
API	Application Program Interface
AQA	Algorithm Quality Assurance
AQAHW	Algorithm Quality Assurance Hardware (Configuration Item)
AQUA	PM-1 Satellite (AIRS, AMSR-E, AMSU, CERES, HSB, MODIS)
AR	Action Request
ARS	Action Request System (by Remedy)

AS	Administration Stations
ASCII	American Standard Code for Information Interchange
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
ATM	Asynchronous Transfer Mode
AURA	NASA mission to study the earth's ozone, air quality and climate (formerly the CHEM mission)

## **B**

BCP	Bulk Copy Program Bulk Copy Procedure
BDS	Bulk Data Server
BLM	Baseline Manager

## **C**

CAD	Computer Aided Design
CCB	Change Control Board (Raytheon Convention) Configuration Control Board (NASA Convention)
CCDI	ClearCase DDTS Integration
CCR	Configuration Change Request
CDE	Common Desktop Environment
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CD-ROM	Compact Disk - Read Only Memory
CDS	Cell Directory Service
CFG	Configuration File
CGI	Common Gateway Interface
CHUI	Character-based User Interface
CI	Configuration Item
CLS	Client Subsystem
CM	Configuration Management

CMI	Cryptographic Management Interface
CMP	Configuration Management Plan
CN	Change Notice
CO	Contracting Officer
COTS	Commercial Off the Shelf (Software or Hardware)
CPF	Calibration Parameter File
CPU	Central Processing Unit
CRM	Change Request Manager
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSMS	Communications and Systems Management Segment (ECS)
CSS	Communications Subsystem

## **D**

DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System
DAO	Data Assimilation Office
DAP	Delivered Algorithm Package
DAR	Data Acquisition Request
DAS	Dual Attached Station
DB	Database
DBMS	Database Management System
DCCI	Distributed Computing Configuration Item
DCN	Document Change Notice
DDICT	Data Dictionary (Computer Software Configuration Item)
DDIST	Data Distribution (Computer Software Configuration Item)
DDR	Detailed Design Review
	Data Delivery Record (same as a Product Delivery Record)
DDT	DAAC Distribution Technician
DDTS	Distributed Defect Tracking System (COTS)

DEM	Digital Elevation Model
DESKT	Desktop (Computer Software Configuration Item)
DEV	Custom Developed Code
DFS	Distributed File System
DID	Data Item Description
DIPHW	Distribution and Ingest Peripheral Hardware Configuration Item
DLL	Dynamic Link Library
DLT	Digital Linear Tape
DM	Data Management
DMGHW	Data Management Hardware (Configuration Item)
DMS	Data Management Subsystem
DNS	Domain Name Service
DOF	Distributed Object Framework
DORRAN	Distributed Ordering, Researching, Reporting, and Accounting Network (At EDC)
DP	Data Provider
DPL	Data Pool Subsystem
DPR	Data Processing Request
DPRID	Data Processing Request Identifier
DPREP	Data Pre-Processing
DPS	Data Processing Subsystem
DR	Data Repository
DRPHW	Data Repository Hardware (Configuration Item)
DSC	Development Solution for the C programming language
DSS	Data Server Subsystem
DTF	Sony DTF Tape cartridge system (replacement for the D3 tape cartridge system)
DTS	Distributed Time Service

## **E**

ECHO	ECS Clearing House
------	--------------------

ECN	Engineering Change Notice
ECS	Earth Observing System Data and Information Core System
EDC	Earth Resource Observation System (EROS) Data Center
EDF	ECS Development Facility
EDG	EOS Data Gateway
EDHS	ECS Data Handling System
EDN	Expedited Data Set Notification
EDOS	Earth Observing System Data and Operations System
EDR	Expedited Data Set Request
EDS	Expedited Data Set
EC	Error conditions (in tickets)
EGS	EOS Ground System
EISA	Enhanced Industry Standard Architecture
E-mail	Electronic Mail (also Email, e-mail, and email)
EMASS	E-Systems Modular Automated Storage Systems
EMOS	ECS Mission Operations Segment (formerly FOS)
EMS <sub>n</sub>	EOSDIS Mission Support network
EOC	Earth Observing System Operations Center
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EROS	Earth Resource Observation System
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth Science Data Type
ESRI	Environmental Systems Research Institute
ETM+	Enhanced Thematic Mapper Plus (Landsat 7)

## **F**

FC	Functional components (capabilities in tickets)
FCAPS	Fault, Configuration, Accountability, Performance, and Security services
FDS	Flight Dynamics System

FH	Fault Handling
FLDB	Fileset Location Database
F&PRS	Functional and Performance Requirements Specification
FSMS	File and Storage Management System
FTP	File Transfer Protocol
FTPD	File Transfer Protocol Daemon

## **G**

GB	gigabyte ( $10^9$ )
Gb	gigabit ( $10^9$ )
GCDIS	Global Change Data and Information System
GCMD	Global Change Master Directory (not developed by ECS)
GDPS	Ground Data Processing System
GDS	Ground Data System (ASTER)
GFE	Government Furnished Equipment
GLAS	Geoscience Laser Altimeter System
GSFC	GODDARD Space Flight Center (NASA facility and DAAC)
GSMS	Ground System Management Subsystem (ASTER)
GTWAY	(Version 0 Interoperability/ASTER) Gateway (Computer Software Configuration Item)
GUI	Graphical User Interface

## **H**

HDF	Hierarchical Data Format
HDF-EOS	an EOS proposed standard for a specialized HDF data format
HMI	Human Machine Interface
HSB	Humidity Sounder for Brazil
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
HWCi	Hardware Configuration Item

## **I**

IAS	Image Assessment System
ICESat	Ice, Cloud and Land Elevation Satellite
ICD	Interface Control Document
ICLHW	Ingest Client Hardware (Configuration Item)
ID	User Identification (or Identifier)
IDG	Infrastructure Development Group
IDL	Interactive Data Language
I/F	Interface
IGS	International Ground Station (Landsat 7)
IHCI	Internetworking hardware configuration item
ILG	Infrastructure Library Group
ILM	Inventory, Logistics, Maintenance (ILM) Manager
IMS	Information Management System (ECS element name)
INGST	Ingest (Computer Software Configuration Item)
INS	Ingest Subsystem
INHCI	Internetworking HWCI
I/O	Input/Output
IP	Internet Protocol International Partner
IRD	Interface Requirements Document
IRR	Incremental Release Review
ISIPS	ICESat Science Investigator-Led Processing System
ISO	International Standards Organization
ISS	Internetworking Subsystem
I&T	Integration and Test

## **J**

JDT	Java Data Acquisition Request Tool (Java DAR Tool)
JESS	Java Earth Science Server



JEST	Java Earth Science Tool
JPL	Jet Propulsion Laboratory (DAAC)

## **K**

KFTP	Kerberos File Transfer Protocol
------	---------------------------------

## **L**

L0 - L4	Level-0 through Level-4 data (ECS)
L0R	Landsat Reformatted Data
LAMS	Landsat 7 Archive Management System
LAN	Local Area Network
Landsat 7	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LFS	Local File System
LPGS	Landsat 7 Level-1 Product Generation System
LZ77	Lempel-Ziv coding

## **M**

M&O	Maintenance and Operations
MB	Megabyte (10 <sup>6</sup> )
Mbps	Megabits Per Second
MCF	Metadata Configuration File
MCI	Management Software Configuration Item (Computer Software Configuration Item)
MHWCI	(System) Management (Subsystem) Hardware Configuration Item
MISR	Multi-Imaging SpectroRadiometer
MLCI	Management Logistics Configuration Item (Computer Software Configuration Item)
MM	Mode Management

MMO	Mission Management Office
MMS	Mode Management Service
M&O	Maintenance and Operations (Staff)
MOC	Mission Operations Center
MODIS	Moderate-Resolution Imaging SpectroRadiometer
MOJO	Message Oriented middleware of JEST (JDT) Objects
MOPITT	Measurements of Pollution in the Troposphere
MP	Message Passing
MS	Mass Storage
	Microsoft
MSCD	Mirror Scan Correction Data (file)
MSS	System Management Subsystem
MTA	LAMS Metadata File
MTMGW	Machine-to-Machine Gateway
MTP	Distribution Product Metadata File Extension (<filename>.MTP)
MTPE	Mission to Planet Earth

## **N**

NASA	National Aeronautics and Space Administration
NCEP	National Centers for Environmental Predictions
NCR	Non-conformance Report
NESDIS	National Environmental Satellite, Data, and Information Service (NOAA)
Netscape	Browser (for user registration and search engine) [Netscape Communicator]
	Mail component for e-mail transfers
NFS	Network File System
NIS	Network Information Service
NMC	National Meteorological Center (located at National Oceanic and Atmospheric Administration - NOAA)
NNTP	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration

NSI	National Aeronautics and Space Administration Science Internet
NSIDC	National Snow and Ice Data Center (DAAC)

## **Q**

ODL	Object Description Language
ODFRM	On-Demand Product Request Form
ODPR	On-Demand Production Request
ODPRM	On-Demand Production Request Manager (PLS)
OEA	OGC-ECHO Adaptor
OEM	Original Equipment Manufacturer
OMS	Order Manager Subsystem
OMSRV	Order Manager Server
OMGUI	Order Manager Graphical User Interface
OOA	Object oriented analysis
OOD	Object oriented design
OODCE	Object Oriented distributed computing environment
OPS CON	Operations Concept
OS	Operating System
OSI	Open Systems Interconnect
OWS	Open Geospatial Consortium (OGC) Web Services

## **P**

PAN	Production Acceptance Notification
PC	Personal Computer
	Performance Constraints (in tickets)
PCD	Payload Correction Data (file)
PCFG	Process Configuration File
PDPS	Planning and Data Processing Subsystems
PDR	Product Delivery Record
PDRD	Product Delivery Record Discrepancy
PDS	Production Data Set

	Product Distribution System (from EDC and integrated into ECS)
PF	Process Framework
PGE	Product Generation Executable
PI	Principal Investigator
PLANG	Production PLANninG (Computer Software Configuration Item)
PLNHW	Planning Hardware (Configuration Item)
PLS	Planning Subsystem
PM-1	EOS Afternoon Equator Crossing Mission (See Aqua); Mission to study the land, oceans and the earth's radiation budget
PMPDR	Physical Media Product Delivery Record
PPDS	Planning and Processing Data Server
PR	Production Request
PRONG	(Data) PROcessiNG (Computer Software Configuration Item)
PSA	Product Specific Attributes
PWB	Planning WorkBench

## **Q**

QA	Quality Assurance
QDS	Quick Look Data Set (Same as Expedited Data Set)

## **R**

RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RCS	Request Communications Support
RDBMS	Relational Database Management System
REL	Release
RFA	Remote File Access
RFC	Request For Comments
RIP	Routing Information Protocol
RMA	Reliability, Maintainability and Availability

RMS	Request Management Services
ROSE	Request Oriented Scheduling Engine
RPC	Remote Procedure Call
RSC	Raytheon Systems Company
RTU	Rights to Use

## S

SAGE III	Stratospheric Aerosol and Gas Experiment III
SAS	Single Attached Station
SATAN	Security Administrator Tool for Analyzing Networks
SBSRV	Subscription Server
SCF	Science Computing Facility
SCLI	SDSRV Command Line Interface
SCSI	Small Computer System Interface
SDE	Software Development Environment
SDF	Software Development Folder
SDP	Science Data Processing
SDPS	Science Data Processing Segment (ECS)
SDPTK	Science Data Processing Toolkit Science Data Processing Toolkit (Computer Software Configuration Item)
SDSRV	Science Data SeRVer (Computer Software Configuration Item)
SGI	Silicon Graphics, Inc.
SIM	Spectral Irradiance Monitor
SIPS	Science Investigator-Led Processing Systems
SMC	System Management Center System Monitoring and Coordination Center
SMP	Symmetric Multi-processor
SMTP	Simple Mail Transport Protocol
SOLSTICE	Solar Stellar Irradiance Comparison Experiment
SORCE	Solar Radiation and Climate Experiment

SPARC	Single Processor Architecture
SPRHW	Science Processing Hardware (Configuration Item)
SQL	Structured Query Language
SQS	Spatial Query Server
SRF	Server Request Framework
SSAP	Science Software Archive Package
SSIT	Science Software Integration and Test
SSS	Spatial Subscription Server Subsystem
STK	StorageTek
STMGT	Storage Management (Computer Software Configuration Item)
Sybase	(ECS) COTS database management product (ASE)
SYSLOG	System Log

## **T**

TAR	Tape Archive
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TELNET	Telecommunications Network
TERRA	EOS AM Mission spacecraft 1, morning equator crossing spacecraft series -- ASTER, MISR, MODIS and MOPITT instruments; Mission to study the land, oceans and the earth's radiation budget
TIM	Total Irradiance Monitor
TM	Thematic Mapper (Landsat)
TT	Trouble Ticket

## **U**

UDP	User Datagram Protocol
UML	Unified Modeling Language
UR	Universal Reference
URL	Universal Resource Locator

USGS	U. S. Geological Survey
UUID	Universal Unique Identifier

## **V**

V0	Version Zero (Version 0)
V0 GTWAY	Version Zero Gateway (V0 GTWAY CSCI)
V0 ODL	Version 0 Object Description Language
VT	Virtual terminal

## **W**

WAN	Wide Area Network
WKBCH	WorkBenCH (Computer Software Configuration Item)
WKSHW	Working Storage Hardware Configuration Item
WRS	Worldwide Reference System
WS	Working Storage
WWW	World Wide Web

## **X**

xAR	x Acquisition Request (where x is any kind of or generic acquisition request)
XBDS	Bulk Data Service Protocol
XDR	External Data Representation
XFS	Extended File System